# COMPARISON OF PREDICTOR-CORRECTOR METHODS

Romeo Liwanag Malig

# NAVAL POSTGRADUATE SCHOOL
## Monterey, California

# THESIS

COMPARISON OF PREDICTOR-CORRECTOR METHODS

by

Romeo Liwanag Malig

Thesis Advisor: U. R. Kodres

June 1972

# Comparison of Predictor-Corrector Methods

by

Romeo Liwanag Malig
Lieutenant Junior Grade, Philippine Navy
B.S., Philippine Military Academy, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL

June 1972

## ABSTRACT

The aim of this paper is to provide convenient predictor-corrector (P-C) methods for obtaining accurate numerical solution at a minimum cost to first order ordinary differential equations (ODE). In pursuing this goal, a unified development of the most popular and efficient P-C methods is presented, which includes derivation of formulas and analysis of error propagation and numerical stability. Each method is then coded and programmed using the Fortran language. Comparative analysis of the different P-C methods include both theoretical and numerical results. The numerical results were obtained by subjecting each method to a wide variety of test ODE, using a maximum of two corrector applications and a uniform series of step size values. By systematic comparison of the performance of each P-C method the most convenient P-C sets in terms of accuracy and minimum cost are established.

TABLE OF CONTENTS

I. INTRODUCTION ---------- 10

II. NATURE OF THE PROBLEM ---------- 13

A. THE NUMERICAL PROBLEM AND NOMENCLATURE ---------- 13

III. DERIVATION OF PREDICTOR-CORRECTOR EQUATIONS ---------- 17

A. BACKWARD DIFFERENCE OPERATOR ---------- 17

B. NEWTON BACKWARD DIFFERENCE FORMULA ---------- 17

C. INTEGRATION FORMULAS ---------- 18

IV. PREDICTOR-CORRECTOR METHODS ---------- 28

A. DEFINITION ---------- 28

B. A SIMPLE PREDICTOR-CORRECTOR SET ---------- 30

C. MODIFIED PREDICTOR-CORRECTOR SET ---------- 30

D. HAMMING MODIFIED PREDICTOR-CORRECTOR SET ---------- 34

E. P-C SETS CONSIDERED IN THE NUMERICAL EXPERIMENTS ---------- 36

V. NUMERICAL STABILITY OF PREDICTOR-CORRECTOR METHODS ---------- 40

A. PROPAGATION OF ERROR IN A ONE-STEP METHOD ---------- 41

B. PROPAGATION OF ERROR IN THE MULTI-STEP METHOD ---------- 46

C. STABILITY ANALYSIS OF A ONE-STEP METHOD ---------- 52

D. STABILITY ANALYSIS OF A MULTI-STEP METHOD ---------- 53

VI. CONVERGENCE OF THE CORRECTOR IN THE P-C METHODS ---------- 59

VII. INFLUENCE OF THE PROPAGATED ERROR ---------- 62

VIII. STABILITY BOUNDS OF P-C SETS ------------------ 65

A. EXPERIMENTAL PROCEDURE -------------------- 69

B. PREDICTED STABILITY CHARACTERISTIC OF THE P-C SETS ---------------------------- 72

C. NUMERICAL RESULTS FOR REAL NEGATIVE STABILITY LIMITS --------------------------- 80

1. Euler (P-C-I) ------------------------- 81

2. Milne (P-C-II) ------------------------ 83

3. Nystrom (P-C-III) --------------------- 83

4. Hermite (P-C-IV) ---------------------- 86

5. Hamming (P-C-V) ----------------------- 86

6. Second Order Adams (P-C-VI) ----------- 89

7. Third Order Adams (P-C-VII) ----------- 89

8. Fourth Order Adams (P-C-VIII) --------- 92

D. NUMERICAL RESULTS FOR RELATIVE STABILITY --- 94

IX. NUMERICAL EXPERIMENTS ------------------------ 106

A. ODE I ------------------------------------- 107

B. ODE II ------------------------------------ 114

C. ODE III ----------------------------------- 118

D. ODE IV ------------------------------------ 118

E. ODE V ------------------------------------- 124

F. ODE VI ------------------------------------ 129

G. ODE VII ----------------------------------- 132

H. ODE VIII ---------------------------------- 137

I. ODE IX ------------------------------------ 142

J. ODE X ------------------------------------- 142

K. SUMMARY ----------------------------------- 151

X. CONCLUSIONS------------------------------------ 158

XI. EXTENSIONS ---------------------------------------- 161

FLOW CHARTS ------------------------------------------- 164

COMPUTER PROGRAM -------------------------------------- 170

LIST OF REFERENCES ------------------------------------ 201

INITIAL DISTRIBUTION LIST ----------------------------- 202

DD FORM 1473 ------------------------------------------ 203

LIST OF TABLES


1. Influence of Propagated Error (Σ) for y' = -y at x=1.0 ---------- 63

2. Absolute Error (Σ) for Euler (P-C-I) ---------- 82

3. Absolute Error (Σ) for Milne (P-C-II) ---------- 84

4. Absolute Error (Σ) for Nystrom (P-C-III) ---------- 85

5. Absolute Error (Σ) for Hermite (P-C-IV) ---------- 87

6. Absolute Error (Σ) for Hamming (P-C-V) ---------- 88

7. Absolute Error (Σ) for Second Order Adams (P-C-VI) ---------- 90

8. Absolute Error (Σ) for Third Order Adams (P-C-VII) ---------- 91

9. Absolute Error (Σ) for Fourth Order Adams (P-C-VIII) ---------- 93

10. True Solution Values for ODE: y'=y with y(0)=1 ---------- 95

10A. Absolute Error (Σ) for Euler (P-C-I) ---------- 97

10B. Absolute Error (Σ) for Nilne (P-C-II) ---------- 98

10C. Absolute Error (Σ) for Nystrom (P-C-III) ---------- 99

10D. Absolute Error (Σ) for Hermite (P-C-IV) ---------- 100

10E. Absolute Error (Σ) for Hamming (P-C-V) ---------- 101

10F. Absolute Error (Σ) for Second Order Adams (P-C-VI) ---------- 102

10G. Absolute Error (Σ) for Third Order Adams (P-C-VII) ---------- 103

10H. Absolute Error (Σ) for Fourth Order Adams (P-C-VIII) ---------- 104

11. Test ODEs ---------- 108

12. True Solution Values for ODE I ------------------- 109

12a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE I at x = 10.0 ------------------ 110

12b. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE I at h = 0.5 ------------------- 111

12c. Absolute Error for P-C-V, P-C-VI, P-C-VII, and P-C-VIII; ODE I at x = 10.0 ---------------- 113

13. True Solution Values for ODE II ------------------ 115

13a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE II at x = 10.0 ----------------- 116

13b. Absolute Error for P-C-V, P-C-VI, P-C-VII, and P-C-VIII; ODE II at x = 10.0 --------------- 117

14. True Solution Values for ODE III ----------------- 119

14a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE III at x = 10.0 ---------------- 120

14b. Absolute Error for P-C-V, P-C-VI, P-C-VII, and P-C-VIII; ODE III at x = 10.0 -------------- 121

15. True Solution Values for ODE IV ------------------ 122

15a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE IV at x= 10.0 ------------------ 123

15b. Absolute Error for P-C-V, P-C-VI, P-C-VII, and P-C-VIII; ODE IV at x = 10.0 --------------- 125

16. True Solution Values for ODE V ------------------- 126

16a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE V at x = 10.0 ------------------ 127

16b. Absolute Error for P-C-V, P-C-VI, P-C-VII, and P-C-VIII; ODE V at x = 10.0 ---------------- 128

17. True Solution Values for ODE VI ------------------ 130

17a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE VI at x = 10.0 ----------------- 131

17b. Absolute Error for P-C-VI, P-C-VI, P-C-VII, and P-C-VIII; ODE V at x = 10.0 ---------------- 133

18. True Solution Values for ODE VII ----------------- 134

18a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE VII at x = 10.0 ---------------- 135

18b. Absolute Error for P-C-V, P-C-VI, P-C-VII, and P-C-VIII; ODE VII at x = 10.0 -------------- 136

19. True Solution Values for ODE VIII ---------------- 139

19a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE VIII at x = 10.0 --------------- 140

19b. Absolute Error for P-C-V, P-C-VI, P-C-VII, and P-C-VIII; ODE VIII at x = 10.0 ------------- 141

20. True Solution Values for ODE IX ------------------ 143

20a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE IX at x = 10.0 ----------------- 144

20b. Absolute Error for P-C-V, P-C-VI, P-C-VII, and P-C-VIII; ODE IX at x = 10.0 --------------- 145

21. True Solution Values for ODE X ------------------- 148

21a. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE X at x = 10.0 ------------------ 149

21b. Absolute Error for P-C-I, P-C-II, P-C-III, and P-C-IV; ODE X at h = 0.5 ------------------- 150

21c. Absolute Error for P-C-V, P-C-VI, P-C-VII, and P-C-VIII; ODE X at x = 10.0 ---------------- 152

22. Summary of Results for ODE I to ODE IV for $h = 2^{-7}$ to $h = 2^{-1}$ at x = 10.0 ----------------- 153

22a. Summary of Results for ODE V to ODE VIII for $h = 2^{-7}$ to $h = 2^{-1}$ at x = 10.0 ----------------- 154

22b. Summary of Results for ODE IX and ODE X for $h = 2^{-7}$ to $h = 2^{-1}$ at x = 10.0 ----------------- 155

## LIST OF FIGURES

1. Solution Curves of y'=xy ---------------------------- 14

2. Influence of Propagated Error --------------------- 64

3. Characteristic Root for Euler Corrector Formula ------------------------------------------ 74

4. Characteristic Roots for Milne Corrector Formula ------------------------------------------ 76

5. Characteristic Roots for Hamming Corrector Formula ------------------------------------------ 77

# I. INTRODUCTION

A linear first order ordinary differential equation (ODE) can be used as a mathematical model for a variety of phenomena, either physical or non-physical. Examples of such phenomena include the following: heat flow problems (thermodynamics), simple electrical circuits (electrical engineering), force problems (mechanics), rate of bacterial growth (biological science), rate of decomposition of radioactive material (atomic physics), crystallization rate of a chemical compound (chemistry), and rate of population growth (statistics).

Most scientists, engineers or applied mathematicians, who have faced the problem of solving an ordinary differential equation numerically, are probably aware of the multitude of techniques available for such a problem. The abundant literature on the subject of numerical solution of ordinary differential equations is on the one hand, a result of the tremendous variety of actual systems in the physical and biological sciences and engineering disciplines that are described by ordinary differential equations and, on the other hand, a result of the fact that the subject is currently active.

The existence of a large number of methods, each having special advantages, has been a source of confusion as to what methods are best for certain classes of problems. It

is this observation which particularly motivated the writing of this paper; thus it is attempted to bring together the well-known predictor-corrector methods, which form a class of numerical integration methods for solving ordinary differential equations, in a consistent and comprehensive framework.

This delimitation of the study to predictor-corrector methods is not without basis. Further research in the field showed that the predictor-corrector forms are the most efficient among the known integration methods in terms of speed and accuracy. Collectively, as a class of integration methods, the predictor-corrector sets are the best, but individually as a predictor-corrector set, the choice for the best method varies depending on the application. This paper atempts to compare the overall efficiency of all the well-known and most popular predictor-corrector sets by using a standard mode of application, the same series of step size values, and a set of test ODE's with many unusual and interesting features. To introduce the numerical experimentation of the different predictor-corrector methods, a comprehensive analysis of each P-C set starting with its derivation, error propagation, and stability is presented in detail.

Numerical experiments are conducted using the IBM 360/67 digital computer. The tremendous computational capability and speed of this computer offered an indispensable tool in conducting the experiment on a wide variety of test ODEs.

Single precision has seven digit accuracy and double precision has fourteen digit accuracy available for computation.

The paper starts with the description of the nature of ODE initial value problem. Then the various P-C methods for obtaining numerical solutions of the problem are enumerated. To lead into the derivation of formulas a brief review of backward difference operator and the well-known Newton Backward Formula is presented. In turn the different type of P-C methods are discussed followed by the analysis of error propagation and numerical stability. The stability bounds of the P-C methods were established through numerical experimentation. The next step is to subject each P-C method to a wide variety of test ODE. Then systematic comparison of the performance of each P-C method is made. Conclusions are derived and recommendations are made based on the analysis of numerical results obtained. Flow charts and computer programs are attached.

# II. NATURE OF THE PROBLEM

## A. THE NUMERICAL PROBLEM AND NOMENCLATURE

A linear first-order ordinary differential equation (ODE) of the form

$$dy/dx = y'(x) = f(x,y) \tag{1-1}$$

with the initial condition

$$y(x_o) = y_o \tag{1-2}$$

where f(x,y) indicates a differentiable function of the variables x and y is commonly referred to as an initial value problem.

A typical elementary differential equations text presents several general classes of methods for solving a linear first-order ODE. The principal classes of methods are (1) variables-separable, or reduction thereto; (2) exact equations, or reduction thereto; and (3) solution by infinite series. The student is taught to apply the general method that appears best for the solution of the particular ODE. For example, the linear first-order differential equation

$$dy/dx = xy \tag{1-3}$$

can easily be solved by the variables-separable method. This is accomplished by rewriting the equation in the form

$$dy/y = xdx$$

and integrating both sides to obtain

$$\ln y = x^2/2 + c$$

where c is an arbitrary constant of integration. The general solution can then be written as

$$y = c_1 e^{x^2/2}$$

where $c_1 = e^c$. The general solution of such a linear first-order ODE consists of a family curves called the integral curves. The family of integral curves $y = c_1 e^{x^2/2}$ that constitute the solution of equation (1-3) is illustrated in Fig. 1.

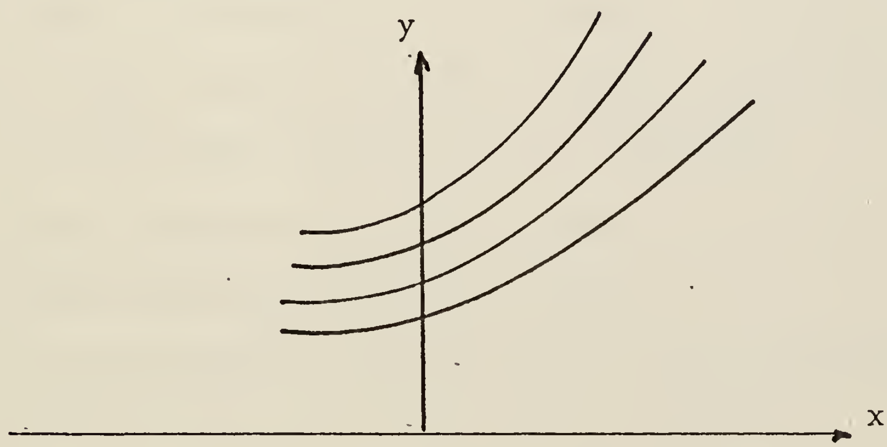



Figure 1. Solution Curves of y' = xy.

For each positive value of $c_1$ a particular member of this family of curves is determined.

A particular solution of equation (1-3) can be determined if a condition on the solution curve is specified. For example, if it is required that the particular solution curve pass through the point (0,1), given by the initial condition

$$y(0) = y_o = 1,$$

then the particular solution $y = e^{x^2/2}$ is obtained since $c_1 = 1$.

However, in real-life problems, many differential equations that are encountered cannot be solved by elementary classical methods. In such instances, one must resort to numerical methods for obtaining one or more particular solutions of the initial-value problems. A multitude of techniques are available for solving such a problem numerically. This paper considers only numerical methods using predictor-corrector equations. Even with this restriction, a large number of methods are in existence. Literature research in this subject, however, reveals that the most popular and efficient methods are the following:

1. Euler Predictor-Corrector Method
2. Milne Predictor-Corrector Method
3. Nystrom Midpoint Predictor-Euler Corrector Method
4. Hermite Predictor-Milne Corrector Method
5. Milne Predictor-Hamming Corrector Method
6. Adams Predictor-Corrector Methods

These methods, each having special advantages, have been a source of confusion as to what methods are best for certain classes of problems. This paper will attempt to present a comparative analysis of these various predictor-corrector methods.

In attempting to compare the various predictor-corrector methods, such questions as the importance of the number of function evaluations, the step size to be used in a numerical

calculation, the computing time required for each method, the influence of truncation and round-off error and the stability of various predictor-corrector modes of computation will be considered.

## III. DERIVATION OF PREDICTOR-CORRECTOR EQUATIONS

### A. BACKWARD DIFFERENCE OPERATOR

To lead into the development of certain equations of major importance in numerically solving ODE, a linear backward difference operator $\nabla$ is defined by the following equations

$$\nabla y_n = y_n - y_{n-1} \tag{2-1}$$

$$\nabla^2 y_n = \nabla y_n - \nabla y_{n-1} = y_n - 2y_{n-1} + y_{n-2}$$

$$\nabla^3 y_n = \nabla^2 y_n - \nabla^2 y_{n-1} = y_n - 3y_{n-1} + 3y_{n-2} - y_{n-3}$$

$$\vdots$$

$$\nabla^q y_r = \nabla^{q-1} y_n - \nabla^{q-1} y_{n-1}$$

In general, these formulas start with a given sequence of the $y(x_n) = y_n$ the ordinates at equally spaced $x_n = x_o + nh$, h is a constant.

### B. NEWTON BACKWARD DIFFERENCE FORMULA

Using these operators the well known Newton backward difference interpolation formula is:

$$y_{n+\alpha} = y_n + \alpha\nabla y_n + \frac{\alpha(\alpha+1)\nabla^2}{2!} y_n + \cdots\cdots \tag{2-2}$$

$$+ \frac{\alpha(\alpha+1)\ldots(\alpha+n-1)}{n!} \nabla^n y_n, \quad \text{where } \alpha = \frac{x_\alpha - x_n}{h}$$

This formula is obtained by fitting a polynomial to the set of points $(x_n, y_n)$ at equally spaced $\{x_n\}$. (Actually the formula is a polyn mial in $\alpha$ because of the change of variable.) Fitting is taken to mean that the polynomial passes through the points $(x_n, y_n)$. Since there are (n+1) points $(y_o, y_1, \ldots, y_n)$, this means that if the solution function y(x) is a polynomial of degree n or less, the formula above will be exact. When y(x) is not a polynomial, or is a polynomial of degree greater than n, the formula will not be exact (except at the points themselves). In other words an error will be made because a finite rather than an infinite series is used. In general form this error will appear as

$$T_\alpha = C_\alpha h^{n+1} y^{[n+1]}(\xi)$$

where C is a function of $\alpha$ and $y^{n+1}(\xi)$ is evaluated at some $\xi$, $x_o < \xi < x_n$. $T_\alpha$ is commonly referred to as the truncation error.

## C. INTEGRATION FORMULAS FOR ODE

Integrating both sides or equation (1-1) between $(x_n, y_n)$ and $(x_{n+1}, y_{n+1})$ there results

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x,y)dx \qquad (2\text{-}3)$$

$$= y_n + \int_{x_n}^{x_{n+1}} y'(x)dx$$

$$= y_n + h \int_0^1 y'(\alpha)d\alpha$$

Attaching the index sequence of n = 0,1,2,... to this equation and noting that $y_o$ is known at $x_o$, (1-2), it becomes apparent that (2-3) can be used recursively to generate $y_1, y_2, y_3, \ldots$ as long as there is some way to evaluate the integral. In other words the solution of the ODE is converted to evaluating an integral.

Derivations of many of the equations used to represent (2-3) now follows:

a) Finite-difference table of backward differences. Assuming that $(x_1,y_1)$, $(x_2,y_2)\ldots(x_n,y_n)$ are given, and computing $y' = f(x_i,y_i)$ for i = 0,1,...n the following table is constructed.

$$
\begin{array}{lllllll}
x_o & y'_o & & & & & \\
 & & \nabla y'_1 & & & & \\
x_1 & y'_1 & & \nabla^2 y'_2 & & & \\
 & & \nabla y'_2 & & \cdot & & \\
x_2 & y'_2 & & \cdot & & \cdot & \\
 & & \cdot & & & & \cdot \\
\cdot & \cdot & & \cdot & & & & \nabla^q y'_n \\
 & & \cdot & & & & \\
\cdot & \cdot & & \cdot & & \cdot & \\
 & & \cdot & & \cdot & & \\
x_{n-1} & y'_{n-1} & & & \nabla^2 y'_n & & \\
 & & & \nabla y'_n & & & \\
x_n & y'_n & & & & &
\end{array}
$$

These values will now be used to continue the solution in computing $y_{n+1}$. In general, by retaining $q^{th}$ differences

in the Newton Backward Formula (NBF) (2-2), the following finite series results.

$$y'_{n+\alpha} = y'_n + \alpha\nabla y'_n + \frac{\alpha(\alpha+1)}{2!}\nabla^2 y'_n + \cdots$$

$$+ \frac{\alpha(\alpha+1)..\alpha(\alpha+q-1)}{q!}\nabla^q y'_n \qquad (2\text{-}4)$$

Using this interpolating polynomial to approximate the integrand in (2-3) [replace $y'(x)$ by $y'_{n+\alpha}$] yields: (complete details of integration are presented in McCalla [Ref. 1])

$$y_{n+1} = y_n + h\sum_{i=0}^{q} a_i\nabla^i y_n, \quad \text{with } a_o = 1$$

where

$$a_i = \int_0^1 \frac{\alpha(\alpha+1)\ldots(\alpha+i-1)}{i!}\,d\alpha, \quad \text{for } i > 0 \qquad (2\text{-}5)$$

Equation (2-4) is in extrapolation form as can be seen from the table of differences, where the end point $x_{n+1}$ is excluded from the interpolating points.

The error term associated with truncating after the $q^{th}$ $\nabla$ is

$$T_\alpha = h^{q+2}\int_0^1 \frac{\alpha(\alpha+1)\ldots(\alpha+q)}{(q+1)!}\,y^{[q+2]}(\xi)\,d\xi$$

But since the coefficient of $y^{[q+2]}$ does not change in sign in $[0,1]$, it is possible to write

$$T_\alpha = a_{q+1}\,h^{q+2}y^{[q+2]}(\xi) \qquad (2\text{-}6)$$

By actually calculating the $a_i$ in (2-5) one arrives at

$$y_{n+1} = y_n + h[1+ \frac{1}{2} \nabla + \frac{5}{12} \nabla^2 + \frac{3}{8} \nabla^3 + \frac{251}{720} \nabla^4 + \ldots ]y'_n$$

Note that by truncating this series at the first difference then (for $q = 0$),

$$y_{n+1} = y_n + hy'_n \quad \text{with } T_\alpha = \frac{1}{2} h^2 y^{[2]}(\xi) \cdot \tag{2-7}$$

which is the well known Euler's formula. Adopting the convention that

$$T_\alpha = T(x,h)$$

since $\alpha$ is a function of x and h. Also

$$T(x,h) = \frac{1}{2} h^2 y^{[2]}(\xi)$$

can be written as

$$T(x,h) = 0(h^2)$$

indicating that the truncation error is of order $h^2$.

Actually (2-3) can be generalized by rewriting it in the form

$$y_{n+1} = y_{n-r} + h \int_{-r}^{1} y'_{n+\alpha} \, d\alpha \tag{2-8}$$

where r is any positive integer. For example, the case $r = 0$ is the one already discussed. Following the same procedure as previously presented, the results for $r = 1,3,5$ are obtained by substituting (2-4) into (2-8) where

$$a_i = \int_{-r}^{1} \frac{\alpha(\alpha+1)\ldots(\alpha+i-1)}{i!} \, d\alpha, \text{ for } i = 1,2,\ldots q.$$

Actually calculating $a_i$ for different values of r there results

$$r = 1; \quad y_{n+1} = y_{n-1} + h[2+0\nabla+ \frac{1}{3}\nabla^2+ \frac{1}{3}\nabla^3+ \frac{29}{90}\nabla^4 + \quad .....]y'_n \tag{2-9}$$

$$r = 3; \quad y_{n+1} = y_{n-3} + h[4-4\nabla+ \frac{8}{3}\nabla^2+0\nabla^3+ \frac{14}{45}\nabla^4 + \quad .....]y'_n \tag{2-10}$$

$$r = 5; \quad y_{n+1} = y_{n-5} + h[6-12\nabla+15\nabla^2-9\nabla^3+ \frac{33}{10}\nabla^4 + 0\nabla^5 + \quad .....]y'_n \tag{2-11}$$

The error term associated with truncation of these formulas is more complicated to calculate because the integrand changes sign in $[-r,1]$. For special cases however, as in the above for $r = 1,3,5$, the coefficient of the $r^{th}$ difference is always zero. Thus it is only necessary to use the $r-1$ differences of $y'$ in order to calculate a result whose truncation error is of the order of $r+2$ in $h$. For $r = q = 1$

$$y_{n+1} = y_{n-1} + 2hy'_n \tag{2-12}$$

with

$$T(x,h) = \frac{h^3}{3} y^{[3]}(\xi) \ .$$

Equation (2-12) is known as the Nystrom Midpoint formula. Similarly for:

$$r=3; \ q=3; \ y_{n+1} = y_{n-3} + 4h[y'_n-\nabla y'_n + \frac{2}{3}\nabla^2 y'_n] \tag{2-13}$$

with

$$T(x,h) = \frac{14}{45} h^5 \; y^{[5]}(\xi)$$

$r = 5;\; q = 5;$

$$y_{n+1} = y_{n-5} + 6h[y_n' - 2\nabla y_n' + \frac{5}{2}\nabla^2 y_n' - \frac{3}{2}\nabla^3 y_n' + \frac{11}{20}\nabla^4 y_n']$$

with

$$T(x,h) = \frac{41}{140} h^7 y^{[7]}(\xi) \qquad .$$

Generally, formulas derived using the extrapolation form of the NBF are referred to as open, explicit or predictor equation because $y_{n+1}$ occurs only on the left hand side of the equation. In other words, $y_{n+1}$ can be calculated directly from the right-hand side values.

b) Extending the table of backward differences by one line to include $x_{n+1}$ as an interpolating point yields

$$\begin{array}{lllll}
x_o & y_o' & & & \\
 & & \nabla y_1' & & \\
x_1 & y_1' & & \nabla^2 y_2' & \\
 & & \cdot & \cdot & \\
\cdot & \cdot & & \cdot & \nabla^q y_n' \\
 & & \cdot & \cdot & \vdots \\
\cdot & \cdot & & & \nabla^q y_{n+1}' \\
 & & \nabla y_n' & & \\
x_n & y_n' & \cdot & \nabla^2 y_{n+1} & \\
 & & \nabla y_{n+1}' & & \\
x_{n+1} & y_{n+1}' & & &
\end{array}$$

NBF now in interpolation form yields a finite series as follows:

$$y'_{n+\alpha} = y'_{n+1} + (\alpha-1)\, y'_{n+1} + \frac{(\alpha-1)(\alpha)}{2!}\, {}^2 y'_{n+1} + \cdots$$

$$+ \frac{(\alpha-1)(\alpha)(\alpha+1)\ldots(\alpha+q-2)}{q!}\, {}^q\, y'_{n+1} \qquad (2\text{-}14)$$

Repeating the procedure used in the extrapolation form (2-4), analagous results for r = 0,1,3 and 5 are

$$r = 0; \quad y_{n+1} = y_n + h[1 - \frac{1}{2}\nabla - \frac{1}{12}\nabla^2 - \frac{1}{24}\nabla^3 - \frac{19}{720}\nabla^4 +$$

$$\ldots]y'_{n+1} \qquad (2\text{-}15)$$

$$r = 1; \quad y_{n+1} = y_{n-1} + h[2 - 2\nabla + \frac{1}{3}\nabla^2 + 0\nabla^3 - \frac{1}{90}\nabla^4$$

$$\ldots]y'_{n+1} \qquad (2\text{-}16)$$

$$r = 3; \quad y_{n+1} = y_{n-3} + h[4 - 8\nabla + \frac{20}{3}\nabla^2 - \frac{8}{3}\nabla^3 + \frac{14}{45}\nabla^4 - 0\nabla^5$$

$$\ldots]y'_{n+1} \qquad (2\text{-}17)$$

$$r = 5; \quad y_{n+1} = y_{n-5} + h[6 - 18\nabla + 27\nabla^2 - 24\nabla^3 + \frac{123}{10}\nabla^4$$

$$- \frac{33}{10}\nabla^5 + \ldots]y'_{n+1} \qquad (2\text{-}18)$$

Some interesting and useful results may be obtained by truncating after a certain number of differences. In the r = 0 case truncation after the first difference (q=1) yields

$$y_{n+1} = y_n + \frac{h}{2}[y'_{n+1} + y'_n] \qquad (2\text{-}19)$$

with

$$T(x,h) = \frac{-h^3}{12} y^{[3]}(\xi)$$

which is called the modified Euler formula. For r = 1, truncating after the second difference (note that the third difference is zero) yields

$$y_{n+1} = y_{n-1} + \frac{h}{3}\left[y'_{n+1} + 4y'_n + y'_{n-1}\right] \tag{2-20}$$

with

$$T(x,h) = \frac{-h^5}{90}\, y^{[5]}(\xi)$$

which is popularly known as Milne's corrector. Formulas resulting from the use of NBF interpolation form are referred to as closed, implicit or corrector equations since $y_{n+1}$ occurs on both sides of the equation. In other words unknown $y_{n+1}$ cannot be calculated directly since it is contained within $y'_{n+1}$.

c) In sections a and b the equations were obtained directly from manipulating interpolation or extrapolation polynomials using NBF. However, another method, called the method of undetermined coefficients, applied to the generalized linear, K-step differential-difference equation with constant coefficients

$$y_{n+1} = \sum_{i=0}^{K} \alpha_i y_{n-i} + h \sum_{i=-1}^{K} \beta_i y'_{n-i} \tag{2-21}$$

will yield all the formulas derived so far and at the same time can be used to obtain all other predictor-corrector equations by suitable choice of the parameters $\alpha_i$ and $\beta_i$. But since the previous formulas derived are sufficient for applications and only the Hermite predictor is of interest

using the present method, the tedious and long process of derivations, which can be found in most numerical analysis books [Refs. 1,2,3], are left out and only the results for the Hermite predictor from [Ref. 4] are presented. There it was found that

$$\alpha_o = -4 \quad \alpha_1 = 5 \quad \alpha_2 = \alpha_3 = 0$$

$$\beta_o = 4 \quad \beta_1 = 2 \quad \beta_{-1} = \beta_2 = \text{---} = \beta_8 = 0$$

yielding

$$y_{n+1} = -4y_n + 5y_{n-1} + h\,[4y'_n + 2y'_{n-1}] \tag{2-22}$$

with

$$T(x,h) = \frac{h^4}{6}\, y^{[4]}(\xi)$$

d) It is of interest to note that the numerical methods developed for solving the initial value problem differ in their requirements of the number of starting values. For example, Euler's formula (2-7) needs only the initial condition $y(x_o) = y_o$ and h to start the solution. Methods that determine $y_{n+1}$, when only one point $(x_n, y_n)$ and step size h are known, are commonly called one-step method.

On the other hand, the Nystrom Midpoint formula (2-12) needs starting values $(x_{n-1}, y_{n-1})$ and $(x_n, y_n)$ to continue the solution. Methods that require step size h and more than one point $(x_n, y_n)$, $(x_{n-1}, y_{n-1})$.... in order to compute $y_{n+1}$ are called multi-step methods.

It is clear then that a multi-step method requires a

one-step method to provide the necessary starting values in its computation.

Literature research showed that the Runge-Kutta method is the best among the available one-step methods. Hence the fourth order Runge-Kutta method based on the Taylor's series expansion truncated after terms of the fourth order (derivation of this method is given in Ref. 3) was used as a starting method for the multi-step methods considered. This has the form

$$y_{n+1} = y_n + \frac{K_1+2K_2+2K_3+K_4}{6}$$

where

$$K_1 = hf(x_n, y_n)$$

$$K_2 = hf(x_n + \frac{h}{2}, y_n + \frac{K_1}{2})$$

$$K_3 = hf(x_n + \frac{h}{2}, y_n + \frac{K2}{2})$$

$$K_4 = hf(x_n + h, y_n + K_3)$$

to solve the initial value problem

$$y' = f(x,y) \quad \text{with} \quad y(x_o) = y_o$$

yielding as many starting values as needed.

# IV. PREDICTOR-CORRECTOR METHODS

## A. DEFINITION

Algorithms that use both an explicit formula and an implicit formula are called predictor-corrector methods. The solution approximation computed by the explicit formula is denoted $y^p_{n+1}$ and is called a predictor. This predictor is then used initially in the right side of an implicit formula that computes a corrector $y^c_{n+1}$. The implicit formula can be repeatedly applied, using $y^c_{n+1}$ from the preceding iteration on the right side and computing a new $y^c_{n+1}$ on the left.

To illustrate the procedure, the Euler P-C algorithm (one-step method predictor) and the Nystrom P-C algorithm (multi-step method) are used.

Given the ODE

$y' = f(x,y)$, with initial solution point $(x_n, y_n)$

1) Euler P-C method uses the equation (2-7) as predictor, and (2-19) as corrector

$$y_{n+1} = y_n + hy'_n \tag{2-23}$$

$$y_{n+1} = y_n + \frac{h}{2}\,[y'_{n+1} + y'_n] \tag{2-24}$$

Since (2-23) requires only one solution point $(x_n, y_n)$ and step size h (assumed to be constant for specific application) which can be chosen arbitrarily (the choice actually affects the stability and error propagation of the method as will

be illustrated later), $y'_n = f(x_n, y_n)$ can be evaluated. Hence (2-23) can be computed yielding the result $y^p_{n+1}$ (p means predicted value). Then the derivative of $y^p_{n+1}$ is computed yielding the result $y'_{n+1}$:

$$y'_{n+1} = f(x_{n+1}, y^p_{n+1}) \ .$$

Next (2-24) is used to compute a corrected value called $y^c_{n+1}$ (c means corrected value)

$$y^c_{n+1} = y_n + \frac{h}{2}\left[y'_{n+1} + y'_n\right] .$$

At this point there are two alternatives. First, the computation can be terminated using $y^c_{n+1}$ as the true approximation value for $y_{n+1}$. Then the computation continues to determine the next solution point $y_{n+2}$ by repeating the entire procedure. Second, the corrected value, $y^c_{n+1}$ of $y_{n+1}$ may not be acceptable in the sense that the equality of both sides of (2-24) is not satisfied to as many digits as may be desired. In other words, the corrector has not converged to the desired accuracy (convergence of the corrector formula is an important topic and discussion of this aspect is deferred to a later section). The next step then is to improve the value $y_{n+1}$ on the left side of (2-24) by taking the derivative of $y^c_{n+1}$, calling the result $y'_{n+1}$, and then calculating an improved value of $y_{n+1}$. This is repeated until convergence occurs to as many digits as desired. This convergence term, designated as EPS, is tested against the absolute difference between the last two approximations.

2) The Nystrom P-C method has (2-12) as predictor and (2-19) as corrector.

$$y_{n+1} = y_{n-1} + 2hy'_n \tag{2-25}$$

$$y_{n+1} = y_n + \frac{h}{2}\left[y'_{n+1} + y'_n\right] \tag{2-26}$$

It is clear that before (2-25) can be evaluated another solution point $(x_{n-1}, y_{n-1})$ is needed in addition to the given solution point $(x_n, y_{n-1})$ of the ODE. Another one step method is therefore needed to provide the additional starting values. The Runge-Kutta method can be used to accomplish this. Once these starting values are known, the same procedure as in Euler P-C method is followed to compute the next point using equation (2-25) as predictor and (2-26) as corrector.

## B. A SIMPLE PREDICTOR-CORRECTOR SET

The predictor-corrector methods discussed so far do not take into account the truncation error incurred in retaining $r^{th}$ differences of the infinite series of the NBF. The application is straightforward, predict then correct, and as such it can be called a simple predictor-corrector set.

## C. MODIFIED PREDICTOR-CORRECTOR SET

The P-C set given by (2-12) and (2-19) has a local truncation error of the same order as the set (2-23), (2-24) ($T(x,h) = 0(h^3)$). Hamming [Ref. 4] has used this feature to expand the method of calculation. The local truncation errors for the predictor (2-12) and corrector (2-19) are

$$T(x,h)p = T_{n+1,p} = \frac{h^3}{3} y^{[3]}(\xi_p) \qquad (2\text{-}27)$$

where

$$x_{n-1} < \xi_p < x_{n+1}$$

and

$$T(x,h)_c = T_{n+1,c} = -\frac{h^3}{12} y^{[3]}(\xi_c) \qquad (2\text{-}28)$$

where

$$x_n < \xi_c < x_{n+1}$$

It follows that the exact value $y(x_{n+1})$, of $y$ at $x_{n+1}$, is given by

$$y(x_{n+1}) = y^p_{n+1} + \frac{h^3}{3} y^{[3]}(\xi_p)$$

$$y(x_{n+1}) = y^c_{n+1} - \frac{h^3}{12} y^{[3]}(\xi_c)$$

From these equations, one obtains

$$y^c_{n+1} - \frac{h^3}{12} y^{[3]}(\xi_c) = y^p_{n+1} + \frac{h^3}{3} y^{[3]}(\xi_p)$$

or

$$y^c_{n+1} - y^p_{n+1} = \frac{h^3}{3} y^{[3]}(\xi_p) + \frac{h^3}{12} y^{[3]}(\xi_c) \qquad (2\text{-}29)$$

Now assuming that

$$y^{[3]}(\xi_p) \simeq y^{[3]}(\xi_c) = y^{[3]}(\xi)$$

Over the interval of interest (i.e., the third derivative does not vary greatly over the interval), (2-29) becomes

$$y^c_{n+1} - y^p_{n+1} = \frac{5}{12} h^3 y^{[3]}(\xi) \qquad (2\text{-}30)$$

where

$$x_{n-1} < \xi < x_{n+1}$$

Even at this point in the development some interesting features can be identified. First one notices that (2-30) can be obtained only in case the order of the predictor equation is equal to the order of the corrector equation. Second, there is now a measure of the local truncation error in terms of $y^c_{n+1}$ and $y^p_{n+1}$ which are explicitly calculated in the P-C algorithm. This may be seen more explicitly by comparing (2-27) with (2-30).

$$y^c_{n+1} - y^p_{n+1} = \frac{5}{12} h^3 y^{3}(\xi) \tag{2-31}$$

$$= \frac{5}{4} \left(\frac{1}{3} h^3 y^{3}(\xi)\right)$$

$$\simeq \frac{5}{4} T_{n+1,p}$$

Similarly comparing (2-28) with (2-30)

$$y^c_{n+1} - y^p_{n+1} \simeq -5\, T_{n+1,c} \tag{2-32}$$

Since the local truncation error can now be estimated, the next question is how to use this information to improve the predicted and corrected values. Considering the predicted value first, it can be seen from (2-31) that

$$T_{n+1,p} = \frac{4}{5} (y^c_{n+1} - y^p_{n+1})$$

Assuming that the local truncation error remains approximately constant over two steps, then

$$T_{n+1,p} = \frac{4}{5}(y^c_{n+1} - y^p_{n+1}) = \frac{4}{5}(y^c_n - y^p_n)$$

The predicted value can then be improved or modified by adding the term $\frac{4}{5}(y^c_n - y^p_n)$ to $y^p_{n+1}$ using only information calculated previously.

The corrector can be modified in essentially the same manner since

$$T_{n+1,c} = -\frac{1}{5}(y^c_{n+1} - y^p_{n+1})$$

can be added to $y^c_{n+1}$ to improve this value. On this basis the overall P-C calculation can be written in the following algorithmic steps:

| | |
|---|---|
| Predict: | $p_{n+1} = y_{n-1} + 2hy'_n$ |
| Modify: | $m_{n+1} = p_{n+1} - \frac{4}{5}(p_n - c_n)$ |
| Reevaluate Derivative: | $m'_{n+1} = f(x_{n+1}, m_{n+1})$ |
| Correct: | $c_{n+1} = y_n + \frac{h}{2}(m'_{n+1} + fn)$ |
| Modify: | $y^c_{n+1} = c_{n+1} + \frac{1}{5}(p_{n+1} - c_{n+1})$ |

Then iterate the corrector to convergence. To simplify the presentation, the symbols $m_{n+1}$, $p_{n+1}$ and $c_{n+1}$ have been used.

The use of the modified predictor-corrector set seems to be very attractive but it must be noted that this method has two limitations. First, it rests on the fact that the original P-C equations have equal-order truncation errors, and second, it neglects round-off error.

## D. HAMMING MODIFIED PREDICTOR-CORRECTOR SET

Hamming used equations (2-10) and (2-16), which are known as the Milne P-C set when both are truncated after the third differences, and he revised the corrector to remove the severe stability problems, which will be seen later.

To develop the revised corrector, Hamming started with the generalized equation (2-21) and obtained

$$y_{n+1} = \alpha_o y_n + \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + h[\beta_{-1} y'_{n+1} + \beta_o y'_n + \beta_1 y'_{n-1}].$$

Using the method of undetermined coefficients [complete derivations can be found in Ref. 4], he obtained

$$\alpha_o = \frac{1}{8}(9-9\alpha_1) \qquad \beta_{-1} = \frac{1}{24}(9-\alpha_1)$$

$$\alpha_1 = \alpha_1 \qquad \beta_o = \frac{1}{12}(9+7\alpha_1)$$

$$\alpha_2 = -\frac{1}{8}(1-\alpha_1) \qquad \beta_1 = \frac{1}{24}(-9+17\alpha_1)$$

with

$$T(x,h) = \frac{(-9+5\alpha_1)}{360} h^5 y^{[5]}(\xi)$$

with $\alpha_1$ as a free parameter. Hamming then considered the values of $\alpha_o, \alpha_2, \ldots \beta_1$ as functions of $\alpha_1$ for $\alpha_1$ = 1, 9/17, 1/9, 0, -1/7, -9/31, and -6/10. After looking at the resulting coefficients in terms of equal magnitude, number of zeros, etc., the case $\alpha_1 = 0$ was chosen as the best value since it yields a greater region of real negative stability and a wide range of relative stability. It can be verified

that the case $\alpha_1 = 1$ yields the Milne's corrector (2-10) truncated after the third difference. With $\alpha_1 = 0$ the Hamming predictor-corrector set is of the form

$$y_{n+1}^{p} = y_{n-3} + \frac{4}{3} h \, [2y'_n - y'_{n-1} + 2y'_{n-2}] \qquad (2\text{-}33)$$

which is (2-10) in the case r = 3, q = 3, of the NBF extrapolation form, with

$$T(x,h) = \frac{13}{45} h^5 y^{[5]}(\xi)$$

and

$$y_{n+1}^{c} = \frac{1}{8} [9y_n - y_{n-2}] + \frac{3}{8} h \, [y'_{n+1} + 2y'_n - y'_{n-1}] \qquad (2\text{-}34)$$

with

$$T(x,h) = \frac{-h^5}{40} y^{[5]}(\xi) \ .$$

Next, using the procedure discussed in the previous section on the modified predictor-corrector set, Hamming develops his modified predictor-corrector algorithm as follows:

Predict: $$p_{n+1} = y_{n-3} + \frac{4}{3} h \, [2y'_n - y'_{n-1} + 2y'_{n-2}] \qquad (2\text{-}35)$$

Modify: $$m_{n+1} = p_{n+1} - \left(\frac{112}{121}\right)(p_n - c_n)$$

Reevaluate Derivative: $$m'_{n+1} = f(x_{n+1}, m_{n+1})$$

Correct: $$c_{n+1} = \frac{1}{8} [9y_n - y_{n-2}] + \frac{3}{8} h \, [m'_{n+1} + 2y'_n - y'_{n-1}]$$

Modify: $$y_{n+1}^{c} = c_{n+1} + \frac{9}{121} [p_{n+1} - c_{n+1}]$$

Then the corrector may be iterated to convergence as desired.

## E. P-C SETS CONSIDERED IN THE NUMERICAL EXPERIMENTS

Having set forth the necessary foundation for the predictor-corrector equations, a list of the P-C sets, which were analyzed and actually applied to the numerical solution of the initial value problem in (1-1) is now summarized. In each case the operator is evaluated in terms of $y'_{n+1}$, $y'_n$, $y'_{n-1}, \ldots$

P-C-I: Euler P-C set.

The predictor is equation (2-7) and the corrector is (2-24),

$$y^p_{n+1} = y_n + hy'_n \; ; \qquad T(x,h) = \frac{1}{2} h^2 y^{[2]}(\xi)$$

$$y^c_{n+1} = y_n + \frac{h}{2}[y'_{n+1} + y'_n] \; ; \qquad T(x,h) = -\frac{1}{12} h^3 y^{[3]}(\xi)$$

P-C-II: Milne P-C set.

Equation (2-10), in the case r = 3, q = 3 of the NBF extrapolation form, is used as the predictor, and the corrector is (2-16) where r = 1, q = 3, of the NBF interpolation form. This yields the equations

$$y^p_{n+1} = y_{n-3} + \frac{4}{3} h[2y'_{n-2} - y'_{n-1} + 2y'_n] \; ; \quad T(x,h) = \frac{14}{45} h^5 y^{[5]}(\xi)$$

$$y^c_{n+1} = y_{n-1} + \frac{h}{3}[y'_{n-1} + 4y'_n + y'_{n+1}] \; ; \qquad T(x,h) = -\frac{h^5}{90} y^{[5]}(\xi)$$

P-C-III: Nystrom Predictor-Euler Corrector Set.

The Predictor is (2-12) and the corrector is (2-24) which yields

$$y^p_{n+1} = y_{n-1} + 2hy'_n \; ; \qquad T(x,h) = \frac{h^3}{3} y^{[3]}(\xi)$$

$$y^c_{n+1} = y_n + \frac{h}{2} [y'_{n+1} + y'_n]; \quad T(x,h) = -\frac{1}{12} h^3 y^{[3]}(\xi)$$

P-C-IV: Hermite Predictor-Milne Corrector.

The predictor used is (2-22) and the corrector is (2-16) yielding

$$y^p_{n+1} = -4y_n + 5y_{n-1} + h [4y'_n + 2y'_{n-1}]$$

$$T(x,h) = \frac{h^4}{6} y^{[4]}(\xi)$$

$$y^c_{n+1} = y_{n-1} + \frac{h}{3} [y'_{n-1} + 4y'_n + y'_{n+1}]$$

$$T(x,h) = \frac{-h^5}{90} y^{[5]}(\xi)$$

P-C-V: Hamming Modified Predictor-Corrector Set.

Equation (2-33) is the predictor and (2-34) is the corrector.

$$y^p_{n+1} = y_{n-3} + \frac{4h}{3} [2y'_n - y'_{n-1} + 2y'_{n-2}]$$

$$T(x,h) = \frac{14}{45} h^5 y^{[5]}(\xi)$$

$$y^c_{n+1} = \frac{1}{8} [9y_n - y_{n-2}] + \frac{3h}{8} [y'_{n+1} + 2y'_n - y'_{n-1}]$$

$$T(x,h) = \frac{-h^5}{40} y^{[5]}(\xi)$$

Adams Form (Adams-Bashforth Predictor-Adams-Moulton Corrector Set).

The A-B formulas are (in the case r = 0) equations of the NBF extrapolation form and the A-M formulas are (in the case r=0) equations of the NBF interpolation form. Three methods are considered: the second, third, and fourth order A-B predictor and A-M corrector.

P-C-VI: Second Order Adams P-C Set.

The predictor and corrector are obtained in the case $q=1$ yielding

$$y^p_{n+1} = y_n + \frac{h}{2}[3y'_n - y'_{n-1}]$$

$$T(x,h) = \frac{5}{12} h^3 y^{[3]}(\xi)$$

$$y^c_{n+1} = y_n + \frac{h}{2}[y'_{n+1} + y'_n]$$

$$T(x,h) = \frac{-h^3}{12} y^{[3]}(\xi)$$

P-C-VII: Third Order Adams P-C Set.

The case $q = 2$ in both the extrapolation and interpolation of the NBF form are used as the predictor and corrector equations respectively, resulting in the formulas

$$y^p_{n+1} = y_n + \frac{h}{12}[23y'_n - 16y'_{n-1} + 5y'_{n-2}]$$

$$T(x,h) = \frac{9}{24} h^4 y^{[4]}(\xi)$$

$$y^c_{n+1} = y_n + \frac{h}{12}[5y'_{n+1} + 8y'_n - y'_{n-1}]$$

$$T(x,h) = \frac{-h^4}{24} y^{[4]}(\xi)$$

P-C-VIII: Fourth Order Adams P-C Set.

The case $q = 3$ in both NBF forms yields the predictor-corrector equations

$$y^p_{n+1} = y_n + \frac{h}{24}[55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}]$$

$$T(x,h) = \frac{251}{720} h^5 y^5 (\xi)$$

$$y^c_{n+1} = y_n + \frac{h}{24} [9y'_{n+1} + 19y'_n - 5y'_{n-1} + y'_{n-2}]$$

$$T(x,h) = \frac{-19}{720} h^5 y^5 (\xi)$$

From the P-C sets above an interesting fact is observed. Namely, P-C-I, P-C-III, and P-C-IV used different predictor equations but have the same corrector equation while P-C-II and P-C-V used the same predictor equation but have different corrector equations. The significance of this observation will be clearly demonstrated later on.

# V. NUMERICAL STABILITY OF PREDICTOR-CORRECTOR METHODS

In the numerical solution of an ODE, a sequence of approximations $y_n$ to the true solution $y(x_n)$ is generated. Roughly speaking, the stability of a numerical method refers to the behavior of the difference or error, $y_n - y(x_n)$, as n becomes large. In order to begin the discussion, the various types of errors which are incurred in numerical integration of (2-4) must be considered.

The errors incurred in a single integration step are of two types:

1. The local truncation error or discretization error - the error introduced by the approximation of the differential equation by a difference equation.

2. Errors due to the deviation of the numerical solution from the exact theoretical solution of the difference equation. Included in this class are round-off errors, due to the inability of evaluating real numbers with infinite precision with the use of computer (i.e., computers usually operate on fixed word length), and errors which are incurred if the difference equation is implicit and is not solved exactly at each step.

If a multi-step method is used, an additional source of error results from the use of an auxiliary method (usually a single-step method, e.g., Runge-Kutta method), to develop the needed starting values for the multi-step method.

## A. PROPAGATION OF ERROR IN A ONE-STEP METHOD

The propagation of error in a one-step method can be analyzed by studying the particular representative one-step method of Euler (2-7).

$$y_{n+1} = y_n + hf(x_n,y_n) \tag{2-36}$$

This can be done by determining the relation of the error at step n+1 to the error at step n. To do this let $Y_n$ denote the true solution of the initial value problem $y'(x) = f(x,y)$ with $y(x_o) = y_o$. Then the total accumulated solution error $e_n$ at step n is defined by

$$\Sigma_n = Y_n - y_n \tag{2-37}$$

The numerical values computed by Euler's algorithm (2-36) satisfy the relation

$$y_{n+1} = y_n + hf(x_n,y_n) - R_{n+1} \tag{2-38}$$

where $R_{n+1}$ denotes the round-off errors resulting from evaluating (2-36). Similarly the true solution values satisfy the relation

$$Y_{n+1} = Y_n + hf(x_n,Y_n) + T_{n+1} \tag{2-39}$$

where $T_{n+1}$ denotes the local truncation error in (2-36).

Subtracting (2-38) from (2-39) yields

$$Y_{n+1} - y_{n+1} = Y_n - y_n + h[f(x_n,Y_n) - f(x_n,y_n)] + E_{n+1} \tag{2-40}$$

where $E_{n+1} = T_{n+1} + R_{n+1}$. Inserting (2-37) in (2-40) yields

$$\Sigma_{n+1} = \Sigma_n + h\,[f(x_n,Y_n) - f(x_n,y_n)] + E_{n+1} \tag{2-41}$$

Applying the mean value theorem to the terms inside the bracket of (2-41), this relation between successive errors can be written as

$$\Sigma_{n+1} = \Sigma_n + h\,[(Y_n - y_n)\ f_y(x_n, \bar{y}_n)] + E_{n+1} \tag{2-42}$$

where $f_y$ denotes $\partial f/\partial y$ and $\bar{y}_n$ lies between $y_n$ and $Y_n$. If $|f_y(x,y)| \leq C$ and $|E_{n+1}| \leq E$ where C and E are both positive constants, the error expression above can be replaced by a related first-order difference equation (Henrici Ref. 5 gives an excellent treatment of the solution of difference equations).

$$e_{n+1} = e_n\,[1+hC] + E \tag{2-43}$$

Under the stated assumptions it follows from (2-42) that

$$|\Sigma_{n+1}| \leq |\Sigma_n|\,[1+hC] + E$$

Now if $|\Sigma_n| \leq e_n$ it follows that

$$|\Sigma_{n+1}| \leq e_n\,[1+hC] + E$$

Therefore, from (2-43)

$$|\Sigma_{n+1}| \leq e_{n+1} \tag{2-44}$$

It follows then from (2-44) that

$|\Sigma_o| \leq e_o \rightarrow |\Sigma_1| \leq e_1 \rightarrow$ ....where the symbol $\rightarrow$ means implies that.

That is, the propagated error is bounded by the solution of the related first-order difference equation. Now since $Y_o - y_o = \Sigma_o = 0$, the condition that $|\Sigma_o| \leq e_o$ will be satisfied by setting $e_o = 0$, which is the initial condition for the difference equation.

Defining $G = 1+hC$, the difference equation (2-43) can be written in the form

$$e_{n+1} = Ge_n + E \quad \text{with initial condition } e_o = 0$$

The solution of this first-order difference equation can be found by successive substitution as follows

$$e_1 = Ge_o + E = E$$

$$e_2 = Ge_1 + E = (G+1)E$$

$$e_3 = Ge_2 + E = (G^2+G+1)E$$

$$\vdots$$

$$e_n = Ge_{n-1} + E = (G^{n-1} + G^{n-2} + \ldots + G+1)E$$

The solution $e_n$ can be seen to be a geometric progression and as such can be written as

$$e_n = \left(\frac{G^n - 1}{G - 1}\right)E \tag{2-45}$$

It follows then that the propagated error in Euler's algorithm (2-4) is bounded by the expression

$$|\Sigma_{n+1}| \leq \left(\frac{(1+hC)^{n+1} - 1}{hC}\right) E$$

We shall illustrate this estimate by a numerical example.

Given the initial value problem

$$y' = f(x,y) = -y \quad \text{with} \quad y(0) = 1$$

we apply Euler's formula (2-36). It is required to express the error at $x_1 = 1$ as a function of h assuming that the round-off error $|R_{n+1}| \leq 10^{-7}$ for single precision (double precision yields $|R_{n+1}| \leq 10^{-14}$) in the IBM 360/67 computer.

Computing

$$C = \max |f_y(x,y)|$$

yields

$$C = 1$$

The truncation error associated with (2-36) is given by

$$T_{n+1} = -\frac{1}{2} h^2 y'' (\xi_n) \quad \text{where} \quad x_n \leq \xi_n \leq x_{n+1}$$

Knowledge of the second derivative $y''(x)$ is needed to evaluate this. Thus

$$y''(x) = \frac{d}{dx} (y'(x)) = \frac{d}{dx} (f(x,y))$$

$$= fx + fy \frac{dy}{dx} .$$

Since $f_x = 0$ and $f_y = -1$, this reduces to

$$y''(x) = y.$$

Using (2-45), one obtains

$$e_n \leqq \frac{(1+h)^n - 1}{h} 10^{-7} + \frac{1}{2} h^2 \max_{0 \leq x \leq 1} |y|$$

To evaluate $\max|y|$, it is seen that the analytical solution of the initial value problem is

$$y = e^{-x} .$$

Then for $x = 1$

$$y = e^{-1} = \frac{1}{e}$$

It is clear that max $|y|$ is attained at $x = 0$; that is

$$\max |y| = 1.$$

Substituting into the original expression yields

$$e_n \leqq \left(10^{-7} + \frac{h^2}{2}\right)\left(\frac{(1+h)^n - 1}{h}\right) \tag{2-46}$$

Rewriting this equation yields

$$e_n \leqq \left(\frac{10^{-7}}{h} + \frac{h}{2}\right)((1+h)^n - 1) \tag{2-47}$$

Thus the propagated error incurred in stepping forward from $x = 0$ to $x = 1$, as a function of h, is bounded by the above expression.

The revised form of the propagated error bound shows the influence the various errors have in the numerical solution of the ODE.

Rewriting (2-47) in the form

$$e_n = \left(\frac{R_{n+1}}{h} + T'_{n+1}\right)((1+h)^n - 1) \tag{2-48}$$

where $R_{n+1}$ is still the round-off error

$T'_{n+1}$ is the truncation error reduced by $0(h)$

and ignoring the contents of the second parenthesis, the equation suggests that as $h \to 0$, the truncation error decreases toward zero whereas the round-off error increases toward infinity. To minimize the error, it is clear that h must be chosen so that the truncation error and the round-off error have equal orders of magnitude. A numerical

experiment was conducted to verify this idea. Finally it can be seen that the accumulated error is not simply equal to the sum of the local truncation error and round-off error but must be computed as given by (2-48) which is the solution of the first order difference equation corresponding to (2-36).

## B. PROPAGATION OF ERROR IN THE MULTI-STEP METHOD

As shown in equation (2-21) the most general representation of the Predictor-corrector methods are of the form

$$y_{n+1} = \sum_{i=0}^{K} \alpha_i y_{n-i} + h \sum_{i=-1}^{K} \beta_i y'_{n-i} \qquad (2\text{-}49)$$

However by confining the study to a form represented by

$$y_{n+1} = y_{n-r} + h \sum_{i=-1}^{K} \beta_i f_{n-i}\ , \quad f_{n-i} = y'_{n-i} \qquad (2\text{-}50)$$

simplicity is obtained without loss of generality. It can be verified that most of the formulas in Section III are in the form (2-50). For example, the case $r = 0$ and $\beta_{-1} = 0$ the Adams-Bashforth formulas are obtained as

$$y_{n+1} = y_n + h \sum_{i=0}^{K} \beta_i f_{n-i}$$

while in the case $r = 0$ and $\beta_{-1} \neq 0$ the Adams-Moulton forms result.

The propagation of error in the multi-step method can then be analyzed by studying equation (2-50). Following the same definition as in the previous section with regard to

$$Y_n,\ y_n,\ \Sigma_n,\ R_{n+1},\ T_{n+1},\ E_{n+1},\ C, \text{ and } E,$$

repetition can be avoided.

The solution values generated by (2-50) satisfy the relation

$$y_{n+1} = y_{n-r} + \beta_{-1}f(x_{n+1},y_{n+1}) + h\sum_{i=0}^{K}\beta_i f(x_{n-i},y_{n-i})$$

$$- R_{n+1}. \tag{2-51}$$

Similarly the solution values $Y_n$ satisfy the relation

$$Y_{n+1} = Y_{n-r} + h\beta_{-1}f(x_{n+1},Y_{n+1}) + h\sum_{i=0}^{K}\beta_i f(x_{n-i},Y_{n-i})$$

$$+ T_{n+1}. \tag{2-52}$$

Subtracting (2-51) from (2-52) yields the equation

$$Y_{n+1} - y_{n+1} = Y_{n-r} - y_{n-r} + h\beta_{-1}(f(x_{n+1},Y_{n+1})$$

$$- f(x_{n+1},y_{n+1})) + h\sum_{i=0}^{K}\beta_i(f(x_{n-i},Y_{n-i})$$

$$- f(x_{n-i}) + T_{n+1} + R_{n+1} \tag{2-53}$$

By application of the mean value theorem to (2-53) and using the definition of $\Sigma_n$ and $E_{n+1}$ one obtains

$$\Sigma_{n+1} = \Sigma_{n-r} + h\beta_{-1}(Y_{n+1}-y_{n+1})fy(x_{n+1},\bar{y}_{n+1}) \tag{2-54}$$

$$+ h\sum_{i=0}^{K}\beta_i(Y_{n-i}-y_{n-i})fy(x_{n-i},\bar{y}_{n-i}) + E_{n+1}$$

Using the definition of C,E, and $\Sigma_n$, (2-54) reduces to

$$\Sigma_{n+1} = \Sigma_{n-r} + h\beta_{-1}\Sigma_{n+1}C + h\sum_{i=0}^{K}\beta_i\ {}_{n-i}C + E$$

or

$$\Sigma_{n+1} - h\beta_{-1}\Sigma_{n+1}C = \Sigma_{n-r} + h\sum_{i=0}^{K}\beta_i\Sigma_{n-i}C + E \qquad (2\text{-}55)$$

The related difference equation to (2-55) can be written as

$$e_{n+1}(1-h|\beta_{-1}|C) = e_{n-r} + hC\sum_{i=0}^{K}|\beta_i|e_{n-i} + E \qquad (2\text{-}56)$$

If $|\Sigma_{n-i}| \leq e_{n-i}$ $(i = 0,1,\ldots K)$ where $K \geq r$ it follows that

$$|\Sigma_{n+1}|(\ 1-h\beta_{-1}C|) \leq |\Sigma_{n-r}| + hC\sum_{i=0}^{K}|\beta_i|\,|\Sigma_{n-i}| + |E|.$$

Thus

$$|\Sigma_{n+1}|(|1-h\beta_{-1}C|) \leq e_{n-r} + hC\sum_{i=0}^{K}|\beta_i|e_{n-i} + |E|.$$

From (2-56) it follows that

$$|\Sigma_{n+1}|(|1-h\beta_{-1}C|) \leq e_{n+1}\ (1-hC|\beta_{-1}|)\ .$$

If $hC|\beta_{-1}| < 1$, then

$$|\Sigma_{n+1}| \leq e_{n+1} \qquad (2\text{-}57)$$

That is, if the magnitude of the propagated error is dominated by the solution of the related difference equation for K+1 successive steps, namely $|\Sigma_i| \leq e_i$ $(i = 0,1,\ldots K)$ then by induction, the same is true for all successive integer values of i. Therefore, a bound for the propagated error in the multi-step method can be determined by obtaining a solution of the related linear, inhomogeneous difference equation of (2.55).

As with ODE, (2-56) has homogeneous and particular solutions such that

$$e_n = e_{nH} + e_{np}$$

The homogeneous solution is given by

$$e_{nH} = (1-hC|\beta_{-1}|)u^{K+1} - U^{K-r} - hC(|\beta_o|u^K + |\beta_1|u^{K-1} + \ldots + |\beta_K|) \tag{2-58}$$

The particular solution can be obtained by assuming that $e_n = -\lambda$, then substituting in (2-56)

$$(1-h|\beta_{-1}|C)(-\lambda) - (-\lambda) - hC(-\lambda)[|\beta_o| + |\beta_1| + \ldots + |\beta_K|] = E$$

$$\lambda[hC|\beta_{-1}| + hC \sum_{i=0}^{K} |\beta_i|] = E$$

$$\lambda hC \sum_{i=-1}^{K} |\beta_i| = E$$

$$\lambda = \frac{E}{hC \sum_{i=-1}^{K} |\beta_i|}$$

The general solution of the difference equation of (2-56) can be written as

$$e_n = d_1 u_1^n + d_2 u_2^n + \ldots + d_{K+1} u_{K+1}^n + \frac{E}{hC \sum_{i=-1}^{K} |\beta_i|} \tag{2-59}$$

where $d_1, d_2, \ldots d_{K+1}$ are determined by the initial starting values and where $u_i (i=1,2,\ldots K+1)$ are the roots of the characteristic equation

$$\left[1-hC|\beta_{-1}|\right]u^{K+1} - u^{K-r} - hC(|\beta_o|u^K + |\beta_1|u^{K-1} + \ldots + |\beta_K|) = 0$$

In what follows the assumption is made that the roots are real and distinct (complex roots and multiple roots are discussed in detail in Ref. 4).

It may be observed that the homogeneous part of the characteristic equation for the accumulated error (2-59) is exactly the same as in the characteristic equation for the original difference equation of (2-50). The significance of this point will be used in the analysis of numerical stability following a numerical example.

To illustrate the previous analysis consider the case where $r = 0$ and $K = 0$ in (2-50) which yields

$$y_{n+1} = y_n + \frac{h}{2} [f_{n+1} + f_n] \, . \tag{2-60}$$

This is the modified Euler formula wherein $y_n$ is given by the initial condition and the needed additional point has been supplied by a predictor formula.

Assuming that $f_y(x,y) = -C$ and that $E_{n+1} = E$, the related difference equation yields

$$(1 - \frac{h}{2} (-C)) \, u_1 - 1 - h(-C) \, \frac{1}{2} = 0$$

which has root

$$u_1 = \frac{(1 - \frac{hC}{2})}{(1 + \frac{hC}{2})} \tag{2-61}$$

Using (2-59) the difference equation solution is then

$$e_n = d_1 \left[ \frac{(1 - \frac{hC}{2})}{(1 + \frac{hC}{2}} \right]^n + \frac{E}{hC \, (\frac{1}{2} + \frac{1}{2})}$$

Analyzing the first term with $d_1$ as constant, it remains to show that if

$$\frac{hC}{2} < 1$$

then the total error $e_n$ decreases with n and convergence is assured. Furthermore if this condition is met then

$$\left(\frac{1 - \frac{hC}{2}}{1 + \frac{hC}{2}}\right) < 1 \tag{2-62}$$

and the solution of the difference equation decreases with increasing n. But the expression on the left of (2-62) is exactly the root of the characteristic equation of (2-60); thus it is clear that if $u_1 < 1$, then the solution of the difference equation decreases with n. This idea can be extended to the general characteristic equation (2-50) whose solution may be written in the form

$$y_n = d_1 u_1^n + d_2 u_2^n + \ldots + d_{K+1} u_{K+1}^n \tag{2-63}$$

It has been shown earlier that the total solution error $|\Sigma_{n+1}| \leq e_{n+1}$ is bounded by the solution of the related difference equation. In turn it was observed that $e_n$ depends on the solution of the characteristic equation of the difference equation. Therefore the problem of numerical stability reduces to the analysis of the roots of the characteristic equation of the related difference equation of (2-63).

## C. STABILITY ANALYSIS OF A ONE-STEP METHOD

Given an ODE

$$y' = Ay \quad \text{with} \quad y(x_o) = y_o \tag{2-64}$$

Assuming that A is constant, the analytic solution is given by

$$y(x) = y_o e^{A(x_n - x_o)}$$

Using Euler formula (2-7), the related difference equation with $y'_n = Ay_n$ is

$$y_{n+1} - (1+Ah)y_n = 0$$

whose characteristic root is

$$u_1 = (1+Ah)$$

Therefore the solution is

$$u_1^n = (1+Ah)^n$$

Since there is only one root, (2-63) reduces to

$$y_n = d_1 u_1^n = d_1 (1+Ah)^n$$

where $d_1 = y_o$. Therefore,

$$y_n = y_o(1+Ah)^n = y_o(1+Ah)^{\frac{x_n - x_o}{n}} \tag{2-65}$$

We recall from calculus that

$$e^x = \lim_{h \to 0} (1+h)^{x/h}$$

and using the fact that A is constant (so that $Ah \to 0$ when $h \to 0$), it can be seen that

$$\lim_{h \to 0} (1+Ah)^{\frac{A(x_n - x_o)}{An}} = e^{A(x_n - x_o)} .$$

It follows that (2-65) reduces to

$$y_n = y_o e^{A(x_n - x_o)} .$$

The method is then (roughly speaking) stable, which means that a sequence of approximations $y_n$ will have the true solution as its limit if the corresponding values of h have zero as their limit. This analysis is quite simple since no extraneous roots represented by $d_2 u_2^n$, $d_3 u_3^n \ldots$, are introduced. The case where these roots are introduced will be discussed thoroughly in the next section. The only error introduced is represented by the particular solution of (2-59), whose behavior has been thoroughly studied in Section V-A as a function of h.

## D. STABILITY ANALYSIS OF A MULTI-STEP METHOD

Given the same problem

$$y' = Ay \quad \text{with} \quad y(x_o) = y_o$$

the corrector equation of Milne's P-C set (P-C-II) is used viz.

$$y_{n+1} = y_{n-1} + \frac{h}{3} \left( y'_{n-1} + 4y'_n + y'_{n+1} \right) . \tag{2-66}$$

Since

$$y'_{n-1} = Ay_{n-1}$$

$$y_n' = Ay_n$$

$$y_{n+1}' = Ay_{n+1}$$

(2-66) has the related difference equation in the simplified form

$$y_{n+1}\left[1 - \left[\frac{Ah}{3}\right]\right] - y_n\left[\frac{4Ah}{3}\right] - y_{n-1}\left[1 + \frac{Ah}{3}\right] = 0.$$

The characteristic equation yields

$$u_1^2\left[1 - \frac{Ah}{3}\right] - u_1\left[\frac{4Ah}{3}\right] - \left[1 + \frac{Ah}{3}\right] = 0. \qquad (2\text{-}67)$$

with roots

$$u_1 = \frac{2Ah \pm\sqrt{9 + 3Ah^2}}{3 - Ah}.$$

Assuming that h is sufficiently small, the binomial theorem can be used to expand the square root and then simplifying, one obtains

$$u_1 = 1 + Ah + 0(h)$$

$$u_2 = -1 + \frac{1}{3}Ah + 0(h)$$

Since $n = (x_n - x_o)/h$, the solution of (2-67) can be written as

$$y_n = d_1(1+Ah+0(h))^{(x_n-x_o)/h} + d_2(-1)^n(1 - \frac{1}{3}Ah + 0(h))^{\frac{(x_n-x_o)}{h}}$$

As h tends to zero, this solution tends to

$$y_n = d_1 e^{A(x_n-x_o)} + d_2(-1)^n e^{-\frac{1}{3}A(x_n-x_o)} \qquad (2\text{-}68)$$

Since $d_1$ is given by the initial condition and $d_2$ is assumed to have been supplied by an auxiliary one step method, (2-68) can be completely determined. It is clear that there is an extraneous root introduced as a result of the use of a $2^{nd}$ order difference equation to represent a first-order differential equation. The root $u_2$ is then called spurious, parasitic, or extraneous and has no relation to the exact solution of the differential equation but, nevertheless, is unavoidable. This is clearly seen if it is assumed that $d_2 = 0$, in which case (2-68) reduces to

$$y_n = y_o \, e^{A(x_n - x_o)}$$

This is indeed the true solution of the ODE. But in general $d_2 \neq 0$, so the behavior of this extraneous root remains to be studied, since it affects the total solution. This could be done proving that, if $A > 0$, the parasitic solution

$$u_2^n = (-1)^n \, e^{-\frac{A}{3}(x_n - x_o)}$$

tneds to zero exponentially as $x_n$ increases. Then the effect of the spurious solution becomes negligible as $x_n$ increases while the principal solution

$$u_1^n = e^{A(x_n - x_o)}$$

tends toward the true solution values $y(x_n)$. However if $A < 0$ the spurious solution increases exponentially in magnitude and alternates in sign as the step-by-step calculation progresses, while the principal root $u_1$ tends to one.

Moreover, at each stage of the calculation, rounding errors will introduce new spurious terms of the same type. Since these extraneous roots have no relation to the exact solution, as they become large, they will dominate the results, thus invalidating the numerical solution. This situation is referred to as numerical instability.

The same analysis can be generalized by going back to equation (2-63)

$$y_n = d_1u_1^n + d_2u_2^n + \ldots + d_{K+1}u_{K+1}^n \qquad (2\text{-}69)$$

It is clear then that the principal root is $d_1u_1^n$ and $d_2u_2^m$, $d_3u_3^n, \ldots d_{K+1}u_{K+1}^n$ are extraneous roots that are introduced as a result of representing a first order differential equation by a $(K+1)^{th}$ order difference equation. By analyzing the principal root $u_1$ which has the form

$$u_1 = 1 + Ah + 0(h)$$

or

$$u_1 = e^{Ah} + o(h)$$

more precise requirements about numerical stability can be determined. Observe that $u_1^n$ approximates $e^{nAh}$ as was already shown.

Comparing (2-69) to the exact solution of $y' = Ay$, viz,

$$y(x_n) = e^{nAh} ,$$

it is clear that as long as $|u_1| > |u_i|$, $i = 2,3,\ldots K+1$, when n increases, the extraneous solution represented by $u_i^n$,

will become small when compared to the principal solution $d_1u_1^n$. In this situation the numerical solution is expected to be stable. However, if $|u_i| > |u_1|$ for any $i = 2,3,\ldots K+1$, $u_i^n$ will become large with respect to $u_1^n$ and the numerical solution component corresponding to $u_i^n$ will dominate the solution. This situation is referred to as numerical instability.

Intuitively then, the numerical solution will be valid if $|u_1| > |u_i|$, $i = 2,3,\ldots K+1$. However, recall from the previous analysis of (2-59) that the characteristic equation (2-59) for the accumulated error $e_{n+1}$, is exactly the same as the characteristic equation (2-63) for the original difference equation of (2-50). For a valid numerical solution, it has been shown that the total error $e_n$ must not grow with n. From (2-59), it was observed that

$$e_{nH} = d_1u_1^n + d_2u_2^n + \ldots + d_{K+1}u_{K+1}^n$$

and is equivalent to the condition that $|u_i| \leq 1$, $i,2,3,\ldots K+1$. It is then possible to define the stability of a method in the following way.

(i) ABSOLUTELY STABLE if $|u_i| \leq 1$, $i = 1,2,\ldots K+1$

(ii) RELATIVELY STABLE if $|u_i| \leq u_1$, $i = 2,3,\ldots K+1$

Absolute stability does not imply relative stability. In other words, a numerical solution may have $|u_i| \leq 1$, $i = 1,2,\ldots K+1$, but $|u_1| < |u_i| \leq 1$, $i = 2,3,\ldots K+1$.

Summarizing then, it can be seen that in the ODE

$$y' = Ay$$

absolute stability and relative stability are both important considerations if $A < 0$ since the exact solution is decreasing with $x_n$. If $A > 0$, the exact solution is growing with $x_n$, so that relative stability is the important consideration. In other words, the numerical solution is valid as long as no component of $u_i^n$ dominates the component corresponding to the principal root.

## VI. CONVERGENCE OF THE CORRECTOR IN THE P-C METHODS

The term convergence has been mentioned earlier in the study of the propagation of the error as a function of h. An analysis of the convergence properties of correctors reveals that this convergence indeed depends on the step size h. A proper selection of h must therefore be made to ensure the convergence of the corrector.

Let $y^j_{n+1}$ denote the $j^{th}$ approximation of the solution value at $x_{n+1}$. Then the corrector formula can be written in the form with $\beta_{-1} \neq 0$:

$$y^{j+1}_{n+1} = y_{n-r} + h\beta_{-1} f(x_{n+1}, y^j_{n+1}) + h \sum_{i=0}^{K} \beta_i f(x_{n-i}, y_{n-i}) \tag{2-70}$$

where j is the iteration index. Note that $y^o_{n+1} = y^p_{n+1}$, and $y^{j+1}_{n+1}$ is the corrector of iterating $y^j_{n+1}$ which in turn is the corrector of (j-1) iteration. If the sequence of successive approximations $y^{j+1}_{n+1}$, generated by iterating the corrector, converges to a limit, denoted by $\overset{*}{y}_{n+1}$, then $\overset{*}{y}_{n+1}$ satisfies the relation

$$\overset{*}{y}_{n+1} = y_{n-r} + \beta_{-1} f(x_{n+1}, \overset{*}{y}_{n+1}) + H \sum_{i=0}^{K} \beta_i f(x_{n-i}, y_{n-i}) \tag{2-71}$$

Subtracting (2-71) from (2-70) yields

$$y^{j+1}_{n+1} - \overset{*}{y}_{n+1} = h\beta_{-1} [f(x_{n+1}, y^j_{n+1}) - f(x_{n+1}, \overset{*}{y}_{n+1})]$$

which by the mean value theorem can be written

$$y_{n+1}^{j+1} - y_{n+1}^{*} = h\beta_{-1}\,[y_{n+1}^{j} - y_{n+1}^{*}]\; f_y(x_{n+1},\bar{y}_{n+1})$$

where $\bar{y}_{n+1}$ lies between $y_{n+1}^{*}$ and $y_{n+1}^{j+1}$. If $|f_y| \leq C$, where C is a positive constant, in the neighborhood of $(x_{n+1},y_{n+1}^{*})$, then

$$|y_{n+1}^{j+1} - y_{n+1}^{*}| \leq h|\beta_{-1}|C|y_{n+1}^{j} - y_{n+1}^{*}|.$$

It follows by induction that

$$|y_{n+1}^{j+1} - y_{n+1}^{*}| \leq [h|\beta_{-1}|C]^{j+1}\; |y_{n+1}^{o} - y_{n+1}^{*}|$$

Therefore, the iteration of the corrector will converge to the limit $y_{n+1}^{*}$ provided that

$$|h\beta_{-1}C| < 1 \tag{2-72}$$

With this expression the conditions for convergence of the various corrector formulas considered can be written down immediately:

| Corrector | $\beta_{-1}$ | Condition for Convergence |
|---|---|---|
| Euler | 1/2 | $hC < 2$ |
| Milne | 1/3 | $hC < 3$ |
| Hamming | 3/8 | $hC < 8/3$ |
| Third-order Adams | 5/12 | $hC < 12/5$ |
| Fourth-order Adams | 3/8 | $hC < 8/3$ |

Both the Nystrom and second order Adams P-C sets used the Euler corrector while Hermite used the Milne corrector and thus are included in the above expressions.

An important point that must be brought out here is that convergence will only assure that the sequence of iterations $y_{n+1}^{j+1}$ will converge to some definite value $y_{n+1}$, but not necessarily to the true solution $y(x_n)$. Stability is still the yardstick of a valid numerical solution though (2-72) provides a good rule of thumb to follow in the proper selection of h. Indeed, later it will be shown experimentally that absolute stability can not be attained beyond the bound for h given by (2-72) since convergence is a necessary but not a sufficient condition for stability of a numerical solution (cf. Ralston [Ref. 6]).

# VII. INFLUENCE OF THE PROPAGATED ERROR

In order to test the idea presented in the previous section dealing with the propagation of error, the numerical example given in Section V-A was run on an IBM 360/67 computer where the round-off err bound is $10^{-7}$ using single precision (SP) and $10^{-14}$ in double precision (DP). Recall that the ODE used was

$$y' = -y \quad \text{with} \quad y(0) = 1$$

and the final error expression using the Euler predictor formula at x = 1 is

$$e_n = \left(\frac{10^{-7}}{h} + \frac{h}{2}\right)\left((1+h)^n - 1\right) \tag{2-73}$$

A series of h values from $h = 10^{-6}$ to h = 1.0 was used. The numerical values of $e_n$ were computed both in single precision and double precision. Table 1 shows the actual numerical data obtained. These data show that for large h to $h \simeq 10^{-3}$, the SP and DP results are essentially identical, but as h decreases the round-off errors tend to increase while the truncation error tends to zero. The effect of the round-off error in DP remains negligible since, in double precision, 14 digit accuracy is obtained. These results confirm the idea that the errors decrease with decreasing values of h to a certain point, beyond which the errors increase. A plot of the data obtained is shown in Figure 2.

TABLE 1

INFLUENCE OF PROPAGATED ERROR ($\Sigma$)
FOR $y'=-y$ AT $x=1.0$

| $h$ | $\Sigma^*_{ESP}$ | $\Sigma_{EDP}$ |
|---|---|---|
| $1\cdot10^{-6}$ | $1.595\cdot10^{-1}$ | $8.763\cdot10^{-7}$ |
| $1\cdot10^{-5}$ | $1.596\cdot10^{-2}$ | $8.593\cdot10^{-6}$ |
| $5\cdot10^{-5}$ | $3.434\cdot10^{-3}$ | $4.295\cdot10^{-5}$ |
| $1\cdot10^{-4}$ | $1.780\cdot10^{-3}$ | $8.590\cdot10^{-5}$ |
| $5\cdot10^{-4}$ | $9.011\cdot10^{-4}$ | $2.147\cdot10^{-4}$ |
| $5\cdot10^{-4}$ | $7.722\cdot10^{-4}$ | $4.294\cdot10^{-4}$ |
| $1\cdot10^{-3}$ | $1.029\cdot10^{-3}$ | $8.584\cdot10^{-4}$ |
| $5\cdot10^{-3}$ | $2.211\cdot10^{-3}$ | $2.143\cdot10^{-3}$ |
| $5\cdot10^{-3}$ | $4.311\cdot10^{-3}$ | $4.278\cdot10^{-3}$ |
| $1\cdot10^{-2}$ | $8.540\cdot10^{-3}$ | $8.524\cdot10^{-3}$ |
| $5\cdot10^{-2}$ | $2.106\cdot10^{-2}$ | $2.106\cdot10^{-2}$ |
| $5\cdot10^{-2}$ | $4.133\cdot10^{-2}$ | $4.133\cdot10^{-2}$ |
| $1\cdot10^{-1}$ | $7.968\cdot10^{-2}$ | $7.968\cdot10^{-2}$ |
| $1\cdot0$ | $5.0\ \cdot10^{-1}$ | $5.0\ \cdot10^{-1}$ |

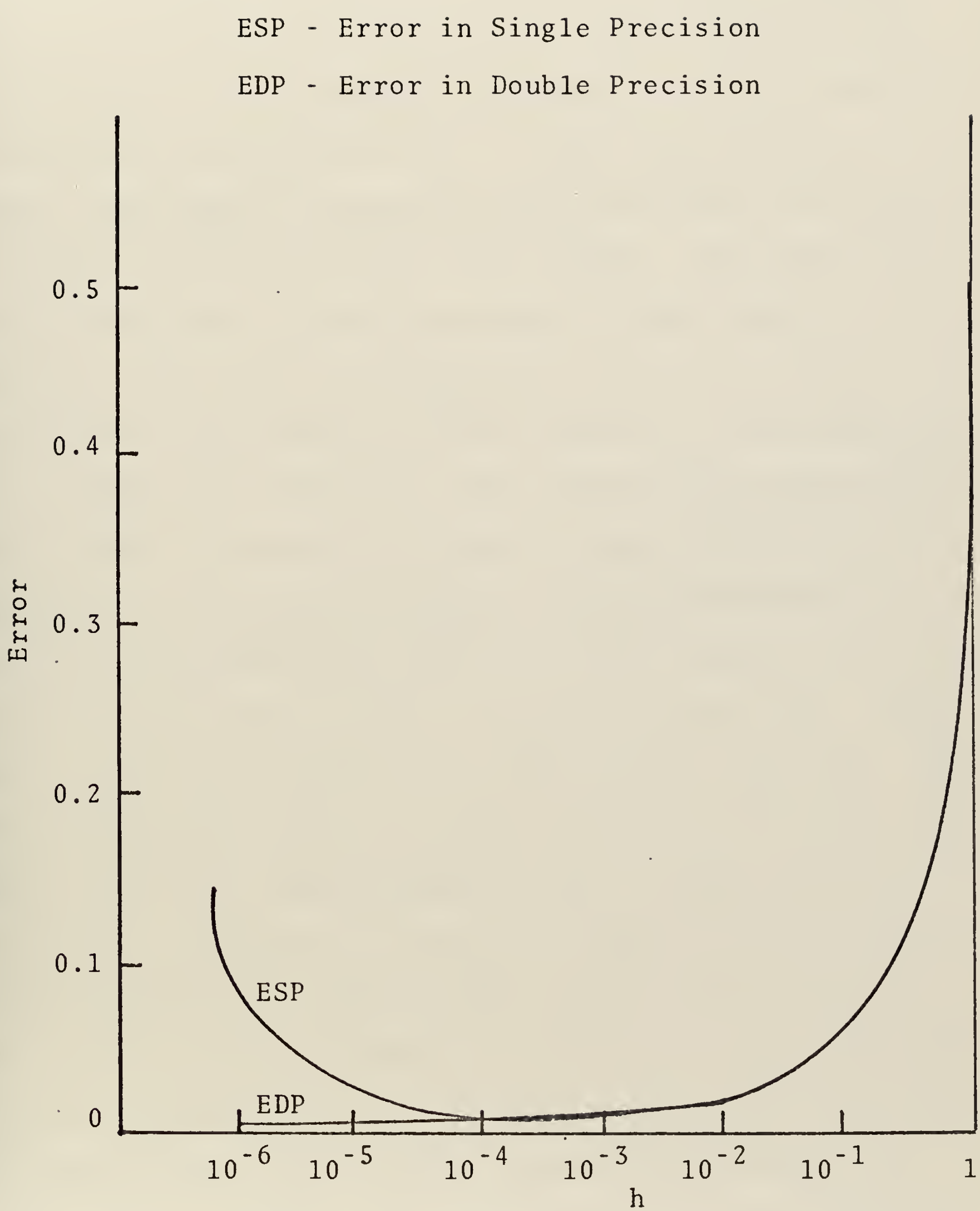



Figure 2. Influence of Propagated Error.

# VIII. STABILITY BOUNDS OF P-C SETS

Prior discussion about the stability of the predictor-corrector methods relates only to the limiting properties of the roots of the characteristic equation as the interval of integration approaches zero. In many applications, however, additional information about the actual size of h is needed to ensure stability of a method. The number of results discussed in published papers on this subject is extensive. But the interesting point observed is that although they all started with the analysis of the roots of the characteristic equations, they varied quite extensively in the mode of the predictor-corrector applications and in the number of P-C sets considered. To be more specific, some published papers are cited. The real negative stability expression used was $h\lambda$ where $\lambda = f_y$ in a single ODE. Chase [Ref. 7] analyzed the Milne P-C mode without iteration. The The real negative stability bounds found were $-0.8 < h\lambda < -0.3$, while on the other hand using the same P-C mode, but now iterated to convergence, resulted in numerical instabilities for all negative $\lambda$. Hamming [Ref. 4] used three modes of his P-C set. First, iterated to convergence, the resulting stability bounds were $-0.5 < h\lambda < 0$; second, with truncation error modification but without iteration the results were $-0.85 < h\lambda < 0$; third, still with truncation error modification but with only two iterations, the results were

$-0.9 < h\lambda < 0$. Lapidus and Seinfeld [Ref. 2] using the PECE and $PE(CE)^s$ methods, where s is the iteration index, published the real negative stability bounds for all available predictor-corrector methods. Their algorithm is quite different from the algorithm used in this paper in that a final derivative evaluation is computed before terminating the computation with or without iteration. The symbol PECE or $PE(CE)^s$ is used to denote their algorithms. The algorithm considered in this paper does not require a final derivative evaluation before termination with or without iteration, and hence can be denoted by $P(EC)^s$.

To establish the real negative stability bounds for the P-C sets considered herein, the experimental procedure used by Chase to obtain the real negative stability bounds for $h\lambda$ will be followed. Chase used the test ODE

$$y' = 100 - 100y \quad \text{with} \quad y(0) = 0$$

Using different values of h, he analyzed the behavior of the error until actual instability occurred. The P-C mode will be different in the sense that a standard application involving two iterations, $P(EC)^2$, will be used for all numerical methods considered. The choice of just two iterations was incluenced first by the published paper of Hull and Creemer [Ref. 8]. They conducted an experiment using the $P(EC)^s$ mode applied to the Adams' methods only. After analyzing several numerical results, they concluded that the best method is S = 2, based on the cost of computation, average accuracy, and stability. Second, the choice of iterations

was influenced by the analysis of the corrector in terms of minimum truncation error. Consider Euler's P-C set (P-C-I), which has the form

Predictor: $y_{n+1} = y_n + hy'_n$

Corrector: $y_{n+1} = y_n + \frac{h}{2} [y'_{n+1} + y'_n]$,

as applied to the ODE $y' = \lambda y$.

Ignoring the corrector, the single characteristic root is obtained from the predictor as follows:

$$y_{n+1} - y_n - h\lambda y_n = 0$$

$$u_1 - (1 + h\lambda) = 0$$

$$u_1 = 1 + h\lambda.$$

Now apply the corrector once. Substituting the predictor into the corrector yields

$$y_{n+1} = (1 + h\lambda + \frac{(h\lambda)^2}{2}) y_n$$

The single characteristic root is

$$u_1 = 1 + h\lambda + \frac{(h\lambda)^2}{2}$$

Now apply the corrector twice obtaining

$$y_{n+1} = y_n + \frac{h}{2} [\lambda(1 + h\lambda + \frac{h^2\lambda^2}{2}) y_n + \lambda y_n]$$

$$y_{n+1} = [1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{\lambda^3 h^3}{4}] y_n$$

The characteristic root is

$$u_1 = 1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{h^3\lambda^3}{4}$$

Continuing this process yields these results

0 corrector: $u_1 = 1 + h\lambda$

1 corrector: $u_1 = 1 + h\lambda + \frac{h^2\lambda^2}{2}$

2 correctors: $u_1 = 1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{h^3\lambda^3}{4}$

3 correctors: $u_1 = 1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{h^3\lambda^3}{4} + \frac{h^4\lambda^4}{8}$

However, the true solution of $y' = \lambda_y$ is given by

$$e^{h\lambda} = 1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{h^3\lambda^3}{6} + \frac{h^4\lambda^4}{24} + 0(h^5)$$

and thus the truncation error $T(x,h)$, given by

$$u_1 - e^{h\lambda}$$

is, first with 0 corrector

$$T(x,h) = (1+h\lambda) - (1+h\lambda + \frac{h^2\lambda^2}{2} + 0(h^3))$$

$$T(x,h) = \frac{-h^2\lambda^2}{2} - 0(h^3)$$

By continuing this procedure with the 1,2,3 correctors and ignoring higher order terms, there results

0 corrector: $T(x,h) = -\lambda^2 h^2/2$

1 corrector: $T(x,h) = -\lambda^3 h^3/6$

2 correctors: $T(x,h) = \lambda^3 h^3/12$

3 correctors: $T(x,h) = \lambda^3 h^3/12$

It can be seen that the use of the corrector more than twice seems to yield little advantage in this case. Thus the mode of P-C application to be used will be represented by the symbol $P(EC)^2$. In this mode of application the stability of the P-C set depends on both the predictor and corrector applications, though more so on the corrector equation. This assertion will be clearly illustrated in the case of the Milne, Hamming and Nystrom P-C sets. Thus, a priori knowledge of the stability of a method can be obtained by analyzing the roots of the characteristic equation of the corrector formula.

## A. EXPERIMENTAL PROCEDURE

Two general algorithms were developed to implement the predictor-corrector methods in Section IV-E. The first algorithm covers seven of the P-C sets, while the second algorithm covers the $8^{th}$ P-C set, i.e., the Hamming modified predictor-corrector set. The flow chart for ALGORITHM 1 is on page 164 and that for ALGORITHM 2 is on page 167. However, the general steps for each algorithm will be outlined here.

Given the necessary starting values, the step size, the convergence term and the range of integration, the iterative predictor-corrector method for computing the numerical solution $y_{n+1}$ is set up as follows:

ALGORITHM 1:

Step 1: Compute the predicted value by the predictor formula.

Step 2: Compute the derivative of the predicted value.

Step 3: Compute the corrected value by the corrector formula.

Step 4: Test for convergence. If convergence is attained go to step 7, otherwise proceed to the next step.

Step 5: Compute the derivative of the corrected value.

Step 6: Compute the new corrected value by the corrector formula, using the value computed in step 3 as the new predicted value.

Step 7: The result is taken as the desired solution point. Advance the integration point by the step size. Return to step 1 to compute the next solution point.

ALGORITHM 2:

Step 1: Compute the predicted value by the predictor formula.

Step 2: Modify the predicted value by adding the truncation error* of the predictor formula.

Step 3: Compute the derivative of the modified predicted value.

Step 4: Compute the corrected value by the corrector formula.

Step 5: Modify the corrected value by adding the truncation error* of the corrector formula.

Step 6: Test for convergence. If convergence is attained go to step 10; otherwise proceed to the next step.

Step 7: Compute the derivative of the modified corrected value.

Step 8: Compute the new corrected value by the corrector formula using the value as computed in step 5 as the new modified predicted value.

---

*Truncation error as used in the formula is not the original truncation error but the modified truncation term expressed as a function of the difference between the corrected and predicted values.

Step 9: Modify the corrected value.

Step 10: The result obtained is accepted as the desired solution point. Advance the integration point by the step size. Return to step 1 to compute the next solution point.

In both algorithms the mode of application is $P(EC)^2$ as can be readily verified. In this scheme, if the corrector is used only once, then the number of function evaluations needed is also one, since it is assumed that the function value for the corrector has already been computed; if the corrector is used twice then two function evaluations are needed. Thus the number of function evaluations is determined by the number of iterations.

The P-C algorithms were coded using the FORTRAN language. The fortran programs for P-C-I to P-C-VIII are listed on pages 170 to 200. In each program the meanings of symbols and parameters are explained through narrative comments.

To determine whether the stability limits have been reached, certain criteria must be followed. Often, when the stability bound is approached through increasing step size h, the method begins to lose accuracy. An inaccurate, though stable, solution may often result in oscillations which appear at first to be due to actual numerical instability. Actual instability, associated with too large a value of h, usually results in an increasing oscillation in the error, or simply, the growth of the error in one direction. The particular error behavior depends on the sign of the parasitic root causing the instability; a negative root causes

oscillations and a positive root causes a uniform error growth.

## B. PREDICTED STABILITY CHARACTERISTIC OF THE P-C SETS

To gain a priori knowledge of the behavior of the stability characteristics of the P-C sets, the roots of the characteristic equations of the corrector formulas will be studied. It must be noted that the behavior of the characteristic roots, though dominating the P-C method used as a set, will be influenced by the behavior of the predictor formula. Since the chosen mode of application, $P(EC)^2$, is quite different from all the other modes of applications published in different papers, the stability limits obtained by several authors could not be used as the stability limits of the P-C set considered here. Thus the actual real negative stability limits will be determined through numerical experiment. However, as noted previously, the analysis of the characteristic roots remains the same, and as such the published results are used and references are cited. This is done solely to conserve space and to avoid the tedious, repetitious process needed to solve the characteristic equations. The procedures have been amply presented and illustrated in detail in the section on numerical stability of predictor-corrector methods.

a) Analysis of the characteristic roots of the corrector formulas.

The P-C sets with common correctors will be grouped together except for the case of the second order Adams'

method which has the same corrector as Euler, and which will be included in Adams' type.

Euler (P-C-I) and Nystrom (P-C-III):

The corrector formula is given by

$$y_{n+1} = y_n + \frac{h}{2} [y'_{n+1} + y'_n]$$

The related difference equation is in the form

$$(1 - \frac{hC}{2})\, y_{n+1} - (1 + \frac{hC}{2})\, y_n = 0, \text{ where } C = f_y(x,y)$$

The single characteristic root is

$$u_1 = \frac{1 + \frac{hC}{2}}{1 - \frac{hC}{2}}$$

It can be seen that if $C < 0$, (i.e., the derivative $\partial f/\partial y$ is negative), then for absolute stability to occur, $|u_1| < 1$ which is satisfied if $hC/2 < 1$. Thus the numerical solution will decrease as does the exact solution If $C > 0$, as long as $hC/2 < 1$ then the numerical solution increases as does the exact solution. Figure 3 exhibits the behavior of the root versus hC as shown by Lapidus and Seinfeld [Ref. 2].

Milne (P-C-II) and Hermite (P-C-IV):

The corrector is of the form

$$y_{n+1} = y_{n-1} + \frac{h}{3} [y'_{n-1} + 4y'_n + y'_{n+1}].$$

The related difference equation is

$$\left[1 - \frac{hC}{3}\right] y_{n+1} - \left[\frac{4hC}{3}\right] y_n - \left[1 + \frac{hC}{3}\right] y_{n-1} = 0.$$

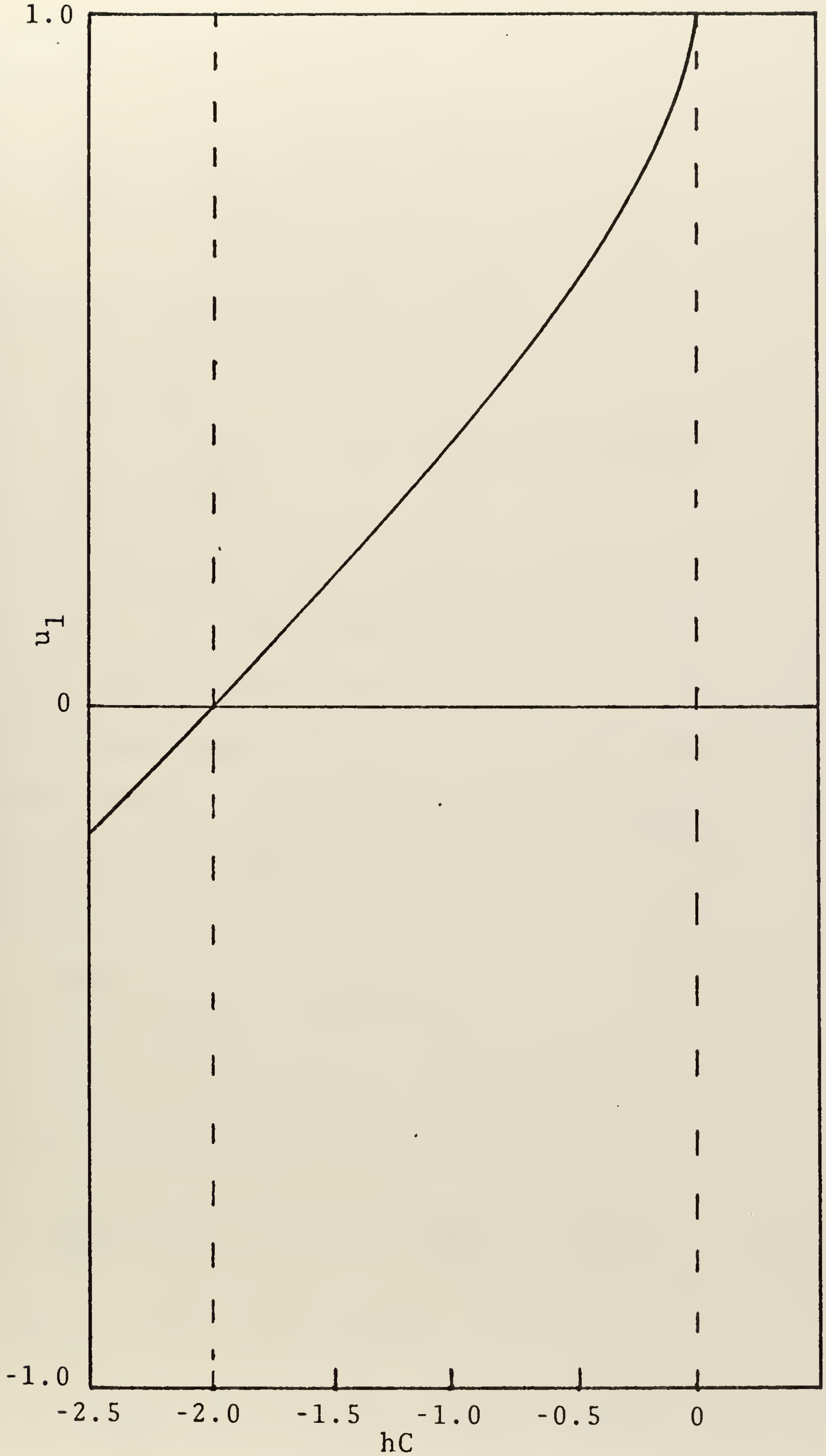



Figure 3. Characteristic Root for Euler Corrector Formula.

The roots of the characteristic equation, as derived in section V-D, are of the form

$$u_1 = 1 + hC + 0(h)$$

$$u_2 = -1 + \frac{1}{3} hC + 0(h)$$

which can be written as

$$u_1 \approx e^{hC}$$

$$u_2 \approx e^{-hC/3}$$

If $C > 0$, $u_1^n$ behaves like the exact solution and $u_2^n$ dies out since, $|u_2| < 1$. When, however, $C < 0$, $u_1^n$ decreases as does the exact solution, but $u_2^n$ increases. Thus the corrector is relatively stable but for $C > 0$, has no real negative stability bounds. Figure 4 shows the behavior of the roots as given by Chase [Ref. 7].

Hamming (P-C-V):

The corrector is given by

$$y_{n+1} = \frac{1}{8} [9y_n - y_{n-2}] + \frac{3h}{8} [y'_{n+1} + 2y'_n - y'_{n-1}]$$

The related difference equation is of the form

$$\left(\frac{3hC}{8} - 1\right) y_{n+1} + \left(\frac{9}{8} + \frac{3hC}{4}\right) y_n - \left(\frac{3hC}{8}\right) y_{n-1} - \frac{1}{8} y_{n-2} = 0$$

The characteristic equation is

$$u^3 \left(\frac{3hC}{8} - 1\right) + u^2 \left(\frac{9}{8} + \frac{3hC}{4}\right) - u\left(\frac{3hC}{8}\right) - \frac{1}{8} = 0$$

Chase [Ref. 7] analyzed the root behavior of this characteristic equation. Figure 5 shows the behavior of these

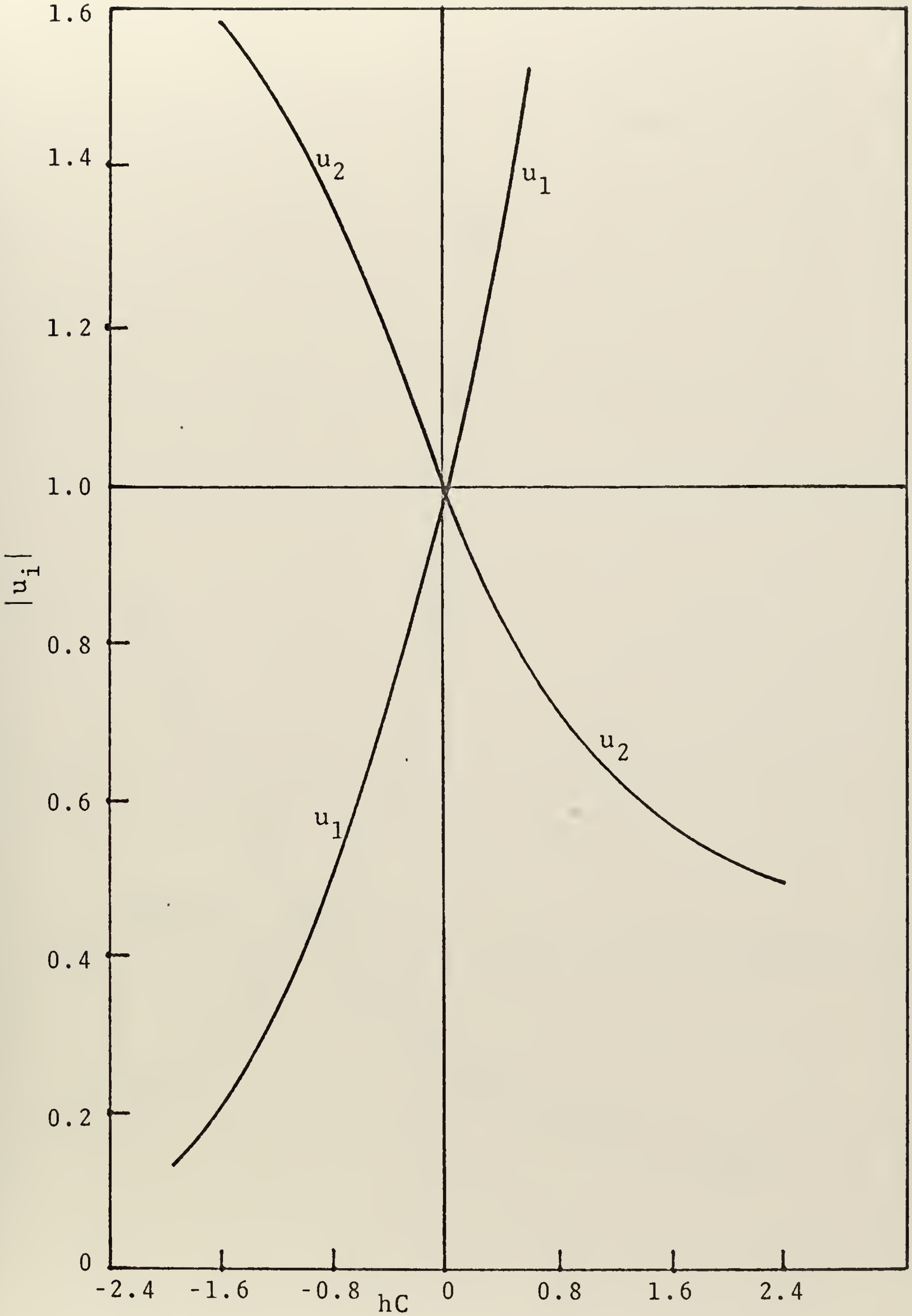



Figure 4. Characteristic Roots for Milne Corrector Formula.

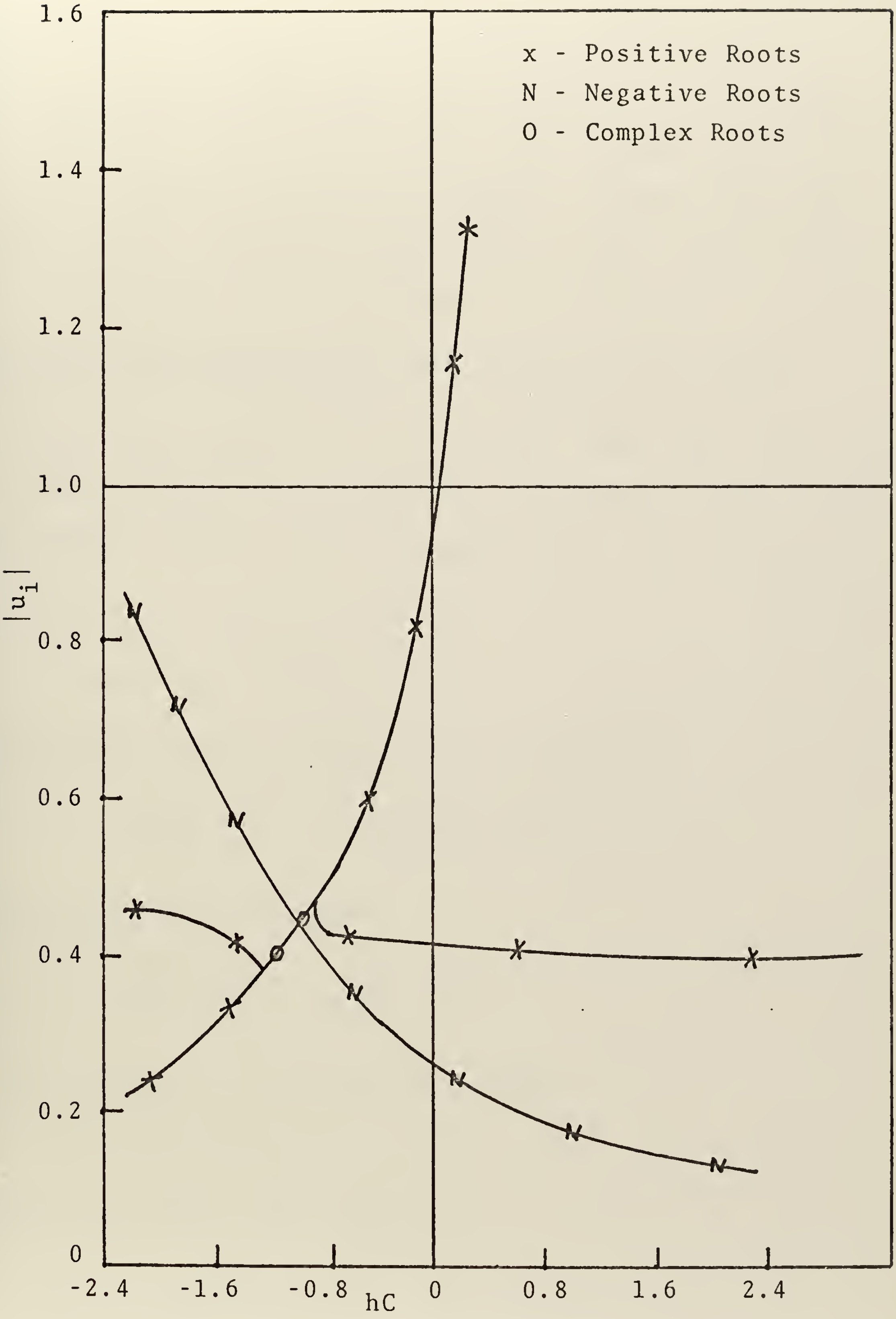



Figure 5. Characteristic Roots for Hamming Corrector Formula.

characteristic roots versus hC. Analyzing the graph of Figure 5, it is evident that the corrector is stable for $-2.4 < hC < 0$. This corrector has also a wide range of relative stability as Chase has shown. In the graph the symbol used for positive roots is an x, that for negative roots is an N, and for the complex roots is an O.

ADAMS-MOULTON CORRECTORS (Second Order (P-C-VI, Third Order (P-C-VII), Fourth Order (P-C-VIII)):

Since these formulas are all of the same type, they will be studied together and extended to the general case of the Adams-Moulton types. For the second order, the corrector is given by

$$y_{n+1} = y_n + \frac{h}{2} [y'_{n+1} + y'_n].$$

The related difference equation is of the form

$$[1 - \frac{hC}{2}] y_{n+1} - [1 + \frac{hC}{2}] y_n = 0$$

with characteristic equation

$$[1 - \frac{hC}{2}] u - [1 + \frac{hC}{2}] = 0$$

In the limit, as $h \to 0$, the root becomes

$$u_1 = 1.$$

For the third order, the corrector is

$$y_{n+1} = y_n + \frac{h}{12} [5y'_{n+1} + 8y'_n - y'_{n-1}]$$

The related difference equation is given by

$$[1 - \frac{5}{12} hC]\, y_{n+1} - [1 + \frac{2}{3} hC]\, y_n + \frac{hC}{12} y_{n-1} = 0$$

with characteristic equation

$$[1 - \frac{5}{12} hC]\, u^2 - [1 + \frac{2}{3} hC]\, u + \frac{hC}{12} = 0$$

In the limiting case $h \to 0$, the roots are

$$u(u-1) = 0$$

$$u_1 = 1 \; ; \; u_2 = 0 \; .$$

For the fourth order, the corrector is

$$y_{n+1} = y_n + \frac{h}{24} [9y'_{n+1} + 19y'_n - 5y'_{n-1} + y'_{n-2}] .$$

The related difference equation is

$$[1 - \frac{3hC}{8}]\, y_{n+1} - [1 + \frac{19}{24} hC] y_n + \frac{5hC}{24} y_{n-1} \frac{-hC}{24} y_{n-2} = 0$$

with characteristic equation

$$[1 - \frac{3}{8} hC] u^3 - [1 + \frac{19}{24} hC] u^2 + \frac{5hC}{24} u - \frac{hC}{24} = 0$$

In the limiting case as $h \to 0$, the roots are

$$u^2(u-1) = 0$$

$$u_1 = 1, \quad u_2 = u_3 = 0$$

Thus, it can be seen that, in each case, the dominant root is 1 while all the other roots (the parasitic ones) lie at the origin when $h = 0$. In the general q-step Adams-Moulton formula the equivalent characteristic equation is

$$u^{q-1}(u-1) = 0 \; .$$

Thus, in every case, the A-M formulas have one root on the unit circle with all others at the origin when $h = 0$. As such they would seem to have excellent stability characteristics no matter what q-step formula is used. When h increases from zero, the parasitic roots move toward the unit circle. However, significantly large values of h can be reached before instability occurs. Crane and Klopfenstein [Ref. 9] analyzed the behavior of the general Adams-Moulton formula and showed that it has stability bounds $-1.3 < hC < 0$. It is interesting to note that, in general, as the order of the Adams corrector increases, accuracy increases but stability decreases. This is the main reason why higher order formulas are not considered (i.e., fifth, sixth, seventh, eighth). The graph of the characteristic roots of the general A-M formulas, as presented by Crane and Klopfstein, was not reproduced since it would give no additional information.

In the foregoing analysis of the characteristic roots, the predicted stability limits served as a guide in the choice of the step size h to be used in the numerical experiment for obtaining the actual real negative stability bounds for each P-C set considered

## C. NUMERICAL RESULTS FOR REAL NEGATIVE STABILITY LIMITS

In order to obtain actual real negative stability bounds of the P-C sets considered in the $P(EC)^2$ mode of application, the ODE

$$y' = -y \quad \text{with} \quad y(0) = 1$$

was chosen, where $f_y(x,y) = C = -1 < 0$.

Each P-C set was run with increasing values of step size h, in small increments, to determine more precisely the actual stability limits. The criteria for actual numerical instability set forth previously, were followed in the analysis of data obtained. The prior knowledge of the roots of the characteristic equation obtained in the previous analysis narrowed the choice of starting values for the step size h for each method. Each method was run in single precision because of the large values of h.

1. Euler (P-C-I)

Since the predicted stability of the corrector is in the range $-2 < hC < 0$, the starting value chosen for h was 1.1, then was increased by 0.1 until instability occurred. Table 2 shows selected results for different values of h. At $h = 1.1$, it is absolutely stable since the error continues to decrease as x increases. The first sign of oscillation was observed at $h = 1.3$ but it is still stable because the error is decreasing. Oscillation becomes evident at $h = 1.6$, but although the solution is inaccurate, it is not yet unstable, because there is no uniformly increasing trend of oscillation. Actual numerical instability occurred at $h = 1.9$ as is clearly shown by the uniformly increasing oscillation. Thus this method is stable within $-1.9 < hC < 0$. As compared to the predicted stability region $-2.0 < hC < 0$ obtained in [Ref. 2] where in the corrector is iterated to covergence, this was expected because of the use of only two iterations.

TABLE 2

ABSOLUTE ERROR ($\Sigma$) FOR EULER (P-C-I)

ODE: $y' = -y$ with $y(0) = 1$
Single Precision

**h = 1.1**

| x | $\Sigma_E$ |
|---|---|
| 1.1 | $-2.239 \cdot 10^{-2}$ |
| 2.2 | $-5.765 \cdot 10^{-2}$ |
| 3.3 | $-2.321 \cdot 10^{-2}$ |
| 4.4 | $-1.610 \cdot 10^{-2}$ |
| 5.5 | $-6.078 \cdot 10^{-3}$ |
| 6.6 | $-3.421 \cdot 10^{-3}$ |
| 7.7 | $-1.266 \cdot 10^{-3}$ |
| 8.8 | $-6.549 \cdot 10^{-4}$ |
| 9.9 | $-2.406 \cdot 10^{-4}$ |

(STABLE)

**h = 1.3**

| x | $\Sigma_E$ |
|---|---|
| 1.3 | $-8.023 \cdot 10^{-2}$ |
| 2.6 | $-1.019 \cdot 10^{-1}$ |
| 3.9 | $-2.748 \cdot 10^{-2}$ |
| 5.2 | $-2.957 \cdot 10^{-2}$ |
| 6.5 | $-1.816 \cdot 10^{-3}$ |
| 7.8 | $-8.301 \cdot 10^{-3}$ |
| 9.1 | $1.629 \cdot 10^{-3}$ |

(STABLE)

**h = 1.6**

| x | $\Sigma_E$ |
|---|---|
| 1.6 | $-2.733 \cdot 10^{-1}$ |
| 3.2 | $-6.44 \cdot 10^{-2}$ |
| 4.8 | $-1.449 \cdot 10^{-1}$ |
| 6.4 | $7.894 \cdot 10^{-2}$ |
| 8.0 | $-1.442 \cdot 10^{-1}$ |
| 9.6 | $1.634 \cdot 10^{-1}$ |

(STABLE BUT INACCURATE)

**h = 1.9**

| x | $\Sigma_E$ |
|---|---|
| 1.9 | $-6.696 \cdot 10^{-1}$ |
| 3.8 | $5.308 \cdot 10^{-1}$ |
| 5.7 | $-1.546$ |
| 7.6 | $3.020$ |
| 9.5 | $-6.363$ |

(UNSTABLE)

2. Milne (P-C-II)

A priori knowledge indicates that this method has no real negative stability, so the starting value of h was chosen as 0.1. Table 3 shows the selected data obtained. Even at the starting value at h = 0.1, the buildup of error is consistent though small. Then at h = 0.4 oscillations become evident and the error tends to increase as x becomes large. At h = 0.7 the same increasing error tendency was observed. These observations, based on actual results, confirmed the predicted instability of this method for $C < 0$. Thus it can be concluded that this method has no real negative stability bounds, as is evident from the actual data presented on Table 3.

3. Nystrom (P-C-III)

The selected starting value is close to the value used in Euler since this method has the same corrector. The program for Nystrom was run starting at h = 1.0 in increments of 0.1 until instability occurred. Table 4 shows the actual data obtained. The result for h = 1.1 to 1.3 show absolute stability. For h = 1.5, the error starts oscillating but not at a uniformly increasing rate. The solution corresponding to this data is inaccurate but stable. This inaccuracy continues up to h = 1.7. For h = 1.8 actual numerical instability has occurred as shown by the increasing oscillation. Thus this method is stable for $-1.8 < hC < 0$. This range of stability is less than that for the Euler method though they have the same corrector equation. This confirmed

TABLE 3

ABSOLUTE ERROR ($\Sigma$) FOR MILNE (P-C-II)

ODE: $y'=-y$ with $y(0)=1$
Single Precision

| h = 0.1 | | h = 0.4 | |
|---|---|---|---|
| x | $\Sigma_M$ | x | $\Sigma_M$ |
| 1.0 | $1 \cdot 10^{-7}$ | 2.4 | $5.17 \cdot 10^{-5}$ |
| 2.0 | $5 \cdot 10^{-7}$ | 3.6 | $-5.86 \cdot 10^{-5}$ |
| 3.0 | $5 \cdot 10^{-7}$ | 4.8 | $1.115 \cdot 10^{-4}$ |
| 4.0 | $5 \cdot 10^{-7}$ | 6.0 | $-1.555 \cdot 10^{-4}$ |
| 5.0 | $6 \cdot 10^{-7}$ | 7.2 | $2.393 \cdot 10^{-4}$ |
| 6.0 | $1 \cdot 10^{-6}$ | 8.4 | $-3.58 \cdot 10^{-4}$ |
| 7.0 | $1.6 \cdot 10^{-6}$ | 9.6 | $5.394 \cdot 10^{-4}$ |
| 8.0 | $2.1 \cdot 10^{-6}$ | (UNSTABLE) | |
| 9.0 | $3.0 \cdot 10^{-6}$ | | |
| 10.0 | $4.2 \cdot 10^{-6}$ | | |
| (UNSTABLE) | | | |

| h = 0.7 | |
|---|---|
| x | $\Sigma_M$ |
| 2.8 | $2.584 \cdot 10^{-4}$ |
| 4.2 | $7.106 \cdot 10^{-4}$ |
| 5.6 | $8.312 \cdot 10^{-4}$ |
| 7.0 | $1.186 \cdot 10^{-3}$ |
| 8.4 | $1.918 \cdot 10^{-3}$ |
| 9.8 | $3.213 \cdot 10^{-3}$ |
| (UNSTABLE) | |

TABLE 4

ABSOLUTE ERROR ($\Sigma$) FOR NYSTROM (P-C-III)

ODE: $y' = -y$ with $y(0)=1$
Single Precision

| h = 1.1 | | h = 1.3 | |
|---|---|---|---|
| x | $\Sigma_N$ | x | $\Sigma_N$ |
| 2.2 | $3.465 \cdot 10^{-2}$ | 2.6 | $5.570 \cdot 10^{-2}$ |
| 3.3 | $4.369 \cdot 10^{-2}$ | 3.9 | $5.976 \cdot 10^{-2}$ |
| 4.4 | $3.043 \cdot 10^{-2}$ | 5.2 | $5.013 \cdot 10^{-2}$ |
| 5.5 | $1.724 \cdot 10^{-2}$ | 6.5 | $1.446 \cdot 10^{-2}$ |
| 6.6 | $7.181 \cdot 10^{-3}$ | 7.8 | $1.914 \cdot 10^{-2}$ |
| 7.7 | $3.062 \cdot 10^{-3}$ | 9.1 | $-1.183 \cdot 10^{-2}$ |
| 8.8 | $4.488 \cdot 10^{-4}$ | (STABLE) | |
| 9.9 | $3.363 \cdot 10^{-4}$ | | |
| (STABLE) | | | |

| h = 1.5 | | h = 1.8 | |
|---|---|---|---|
| x | $\Sigma_N$ | x | $\Sigma_N$ |
| 3.0 | $7.005 \cdot 10^{-2}$ | 3.6 | $-1.865 \cdot 10^{-2}$ |
| 4.5 | $6.538 \cdot 10^{-2}$ | 5.4 | $2.057 \cdot 10^{-1}$ |
| 6.0 | $8.581 \cdot 10^{-2}$ | 7.2 | $-3.134 \cdot 10^{-1}$ |
| 7.5 | $-2.784 \cdot 10^{-2}$ | 9.0 | 1.132 |
| 9.0 | $1.128 \cdot 10^{-1}$ | 10.8 | -3.2066 |
| (STABLE BUT INACCURATE) | | (UNSTABLE) | |

the idea that the stability of the predictor-corrector depends on both the predictor and corrector formula.

4. Hermite (P-C-IV)

This method has the same corrector as the Milne method, hence the choice of starting value is the same, $h = 0.1$, which is then incremented by 0.1 until instability occurred. Table 5 shows the selected actual data obtained. From these results it is evident that this method is unstable since for all values of h the error growth is steadily increasing. This is expected since it was predicted that the corrector has a dominant role in the stability of this method as a P-C set. Thus this method, which uses a Milne corrector, followed the instability behavior of the Milne corrector, within two iterations. Therefore, it can be seen that the Hermite P-C set has no real negative stability bounds.

5. Hamming (P-C-V)

The chosen starting value was $h = 0.5$ with increment 0.1. The predicted stability of the corrector is $-2.4 < hC < 0$. Table 6 shows, that for $h = 0.5$ to $h = 0.6$, the method is absolutely stable. For $h = 0.7$ to $h = 0.8$ the method is stable although the numerical solution becomes inaccurate. For $h = 0.9$ the error grows in one direction thus resulting in actual instability. Thus the method is stable within the range of $-0.9 < hC < 0$. Compared to the predicted stability of the Hamming corrector, given by $-2.4 < hC < 0$, the range of stability is greatly reduced. This is due to the

TABLE 5

ABSOLUTE ERROR ($\Sigma$) FOR HERMITE (P-C-IV)

ODE: $y'=-y$ with $y(0)=1$
Single Precision

| h = 0.2 | | h = 0.5 | |
|---|---|---|---|
| x | $\Sigma_{HE}$ | x | $\Sigma_{HE}$ |
| 1.0 | $-1\cdot10^{-6}$ | 1.0 | $2.820\cdot10^{-4}$ |
| 2.0 | $6\cdot10^{-6}$ | 2.0 | $3.966\cdot10^{-4}$ |
| 3.0 | $-5.1\cdot10^{-6}$ | 3.0 | $6.146\cdot10^{-4}$ |
| 4.0 | $9.1\cdot10^{-6}$ | 4.0 | $9.347\cdot10^{-4}$ |
| 5.0 | $-1.18\cdot10^{-5}$ | 5.0 | $1.401\cdot10^{-3}$ |
| 6.0 | $1.71\cdot10^{-5}$ | 6.0 | $2.088\cdot10^{-3}$ |
| 7.0 | $-2.4\cdot10^{-5}$ | 7.0 | $3.106\cdot10^{-3}$ |
| 8.0 | $3.39\cdot10^{-5}$ | 8.0 | $4.617\cdot10^{-3}$ |
| 9.0 | $-4.78\cdot10^{-5}$ | 9.0 | $6.863\cdot10^{-3}$ |
| 10.0 | $6.74\cdot10^{-5}$ | 10.0 | $1.019\cdot10^{-2}$ |
| (UNSTABLE) | | (UNSTABLE) | |

| h = 0.8 | |
|---|---|
| x | $\Sigma_{HE}$ |
| 1.6 | $2.388\cdot10^{-3}$ |
| 3.2 | $6.331\cdot10^{-3}$ |
| 4.8 | $1.521\cdot10^{-2}$ |
| 6.4 | $3.298\cdot10^{-2}$ |
| 8.0 | $7.033\cdot10^{-2}$ |
| 9.6 | $1.496\cdot10^{-1}$ |
| (UNSTABLE) | |

TABLE 6

ABSOLUTE ERROR ($\Sigma$) FOR HAMMING (P-C-V)

ODE: $y'=-y$ with $y(0)=1$
Single Precision

| h = 0.5 | | h = 0.6 | |
|---|---|---|---|
| x | $\Sigma_{HA}$ | x | $\Sigma_{HA}$ |
| 2.0 | $-2.056\cdot10^{-4}$ | 2.4 | $-3.741\cdot10^{-4}$ |
| 3.0 | $-1.418\cdot10^{-4}$ | 3.6 | $-2.897\cdot10^{-4}$ |
| 4.0 | $-8.06\cdot10^{-5}$ | 4.8 | $-1.689\cdot10^{-4}$ |
| 5.0 | $-4.15\cdot10^{-5}$ | 6.0 | $-9.37\cdot10^{-5}$ |
| 6.0 | $-2.02\cdot10^{-5}$ | 7.2 | $-5.62\cdot10^{-5}$ |
| 7.0 | $-0.6\cdot10^{-6}$ | 8.4 | $-3.78\cdot10^{-5}$ |
| 8.0 | $-4.6\cdot10^{-6}$ | 9.6 | $-2.81\cdot10^{-5}$ |
| 9.0 | $-2.2\cdot10^{-6}$ | (STABLE) | |
| 10.0 | $-1.1\cdot10^{-6}$ | | |
| (STABLE) | | | |

| h = 0.7 | | h = 0.8 | | h = 0.9 | |
|---|---|---|---|---|---|
| x | $\Sigma_{HA}$ | x | $\Sigma_{HA}$ | x | $\Sigma_{HA}$ |
| 2.8 | $-5.709\cdot10^{-4}$ | 3.2 | $-7.459\cdot10^{-4}$ | 3.6 | $-8.327\cdot10^{-4}$ |
| 4.2 | $-5.717\cdot10^{-4}$ | 4.8 | $-1.096\cdot10^{-3}$ | 5.4 | $-1.982\cdot10^{-3}$ |
| 5.6 | $-3.944\cdot10^{-4}$ | 6.4 | $-9.78\cdot10^{-4}$ | 7.2 | $-2.335\cdot10^{-3}$ |
| 7.0 | $-3.047\cdot10^{-4}$ | 8.0 | $-1.121\cdot10^{-3}$ | 9.0 | $-3.801\cdot10^{-3}$ |
| 8.4 | $-2.874\cdot10^{-4}$ | 9.6 | $-1.534\cdot10^{-3}$ | (UNSTABLE) | |
| 9.8 | $-3.001\cdot10^{-4}$ | (STABLE BUT INACCURATE) | | | |
| (STABLE) | | | | | |

effect of the Milne-predictor. But recalling the discussion of the published results, this actual real negative stability bound is exactly the same as that obtained by Hamming using two iterations with truncation error modification. This coincidence is not without basis since the $P(EC)^2$ mode has exactly the same truncation error modification as that of the Hamming method, making the two experimental procedures identical.

6. Second Order Adams (P-C-VI)

The starting value chosen was $h = 0.1$ with an increment of 0.1. Table 7 shows the actual data obtained. For $h = 0.1$ up to $h = 0.3$ the method is stable. But for $h = 0.4$ the method shows actual instability as evidenced by the uniformly increasing error growth. Thus this method is stable within the range $-0.4 < hC < 0$. Compared to the predicted stability of the corrector, which was the same as Euler (P-C-I), with range of stability $-2.0 < hC < 0$, it is evident that the stability is greatly reduced. Again this could be traced to the high instability of the predictor formula used. This is probably the reason why almost all published material on the Adams-Moulton method always starts with the third order and higher predictor. Hull and Creemer [Ref. 8] showed that this method has the largest error among the Adams-type methods.

7. Third Order Adams (P-C-VII)

The choice for h starts at $h = 0.8$ and is then incremented by 0.1. Table 8 shows the selected actual data

TABLE 7

ABSOLUTE ERROR ($\Sigma$) FOR SECOND ORDER ADAMS (P-C-VI)

ODE: $y'=-y$ with $y(0)=1$

Single Precision

| h = 0.1 | | h = 0.2 | |
|---|---|---|---|
| x | $\Sigma_{A2}$ | x | $\Sigma_{A2}$ |
| 1.0 | $3.017\cdot10^{-4}$ | 1.0 | $1.130\cdot10^{-3}$ |
| 2.0 | $2.862\cdot10^{-4}$ | 2.0 | $1.347\cdot10^{-3}$ |
| 3.0 | $2.223\cdot10^{-4}$ | 3.0 | $1.277\cdot10^{-3}$ |
| 4.0 | $1.772\cdot10^{-4}$ | 4.0 | $1.196\cdot10^{-3}$ |
| 5.0 | $1.527\cdot10^{-4}$ | 5.0 | $1.146\cdot10^{-3}$ |
| 6.0 | $1.408\cdot10^{-4}$ | 6.0 | $1.120\cdot10^{-3}$ |
| 7.0 | $1.354\cdot10^{-4}$ | 7.0 | $1.108\cdot10^{-3}$ |
| 8.0 | $1.330\cdot10^{-4}$ | 8.0 | $1.103\cdot10^{-3}$ |
| 9.0 | $1.320\cdot10^{-4}$ | 9.0 | $1.100\cdot10^{-3}$ |
| 10.0 | $1.315\cdot10^{-4}$ | 10.0 | $1.099\cdot10^{-3}$ |
| (STABLE) | | (STABLE) | |
| h = 0.3 | | h = 0.4 | |
| x | $\Sigma_{A2}$ | x | $\Sigma_{A2}$ |
| 1.2 | $2.607\cdot10^{-3}$ | 1.2 | $4.164\cdot10^{-3}$ |
| 2.4 | $3.589\cdot10^{-3}$ | 2.4 | $7.290\cdot10^{-3}$ |
| 3.6 | $3.814\cdot10^{-3}$ | 3.6 | $8.663\cdot10^{-3}$ |
| 4.8 | $3.861\cdot10^{-3}$ | 4.8 | $9.207\cdot10^{-3}$ |
| 6.0 | $3.865\cdot10^{-3}$ | 6.0 | $9.411\cdot10^{-3}$ |
| 7.2 | $3.869\cdot10^{-3}$ | 7.2 | $9.484\cdot10^{-3}$ |
| 8.4 | $3.868\cdot10^{-3}$ | 8.4 | $9.510\cdot10^{-3}$ |
| 9.6 | $3.868\cdot10^{-3}$ | 9.6 | $9.519\cdot10^{-3}$ |
| (STABLE BUT INACCURATE) | | (UNSTABLE) | |

TABLE 8

ABSOLUTE ERROR ($\Sigma$) FOR THIRD ORDER ADAMS (P-C-VII)

ODE: y'=-y with y(0)=1

Single Precision

| h = 0.8 | | h = 0.9 | |
|---|---|---|---|
| x | $\Sigma_{A3}$ | x | $\Sigma_{A3}$ |
| 3.2 | $1.402 \cdot 10^{-2}$ | 3.6 | $1.600 \cdot 10^{-2}$ |
| 4.8 | $7.751 \cdot 10^{-3}$ | 5.4 | $8.272 \cdot 10^{-3}$ |
| 6.4 | $1.968 \cdot 10^{-3}$ | 7.2 | $1.832 \cdot 10^{-3}$ |
| 8.0 | $4.614 \cdot 10^{-4}$ | 9.0 | $7.436 \cdot 10^{-4}$ |
| 9.6 | $4.608 \cdot 10^{-4}$ | (STABLE) | |
| (STABLE) | | | |
| h = 1.2 | | h = 1.3 | |
| x | $\Sigma_{A3}$ | x | $\Sigma_{A3}$ |
| 3.6 | $7.06 \cdot 10^{-5}$ | 3.9 | $-4.736 \cdot 10^{-3}$ |
| 4.8 | $8.773 \cdot 10^{-3}$ | 5.2 | $1.426 \cdot 10^{-3}$ |
| 6.0 | $5.019 \cdot 10^{-3}$ | 6.5 | $1.569 \cdot 10^{-3}$ |
| 7.2 | $5.216 \cdot 10^{-3}$ | 7.8 | $1.869 \cdot 10^{-3}$ |
| 8.4 | $-2.976 \cdot 10^{-3}$ | (UNSTABLE) | |
| 9.6 | $3.050 \cdot 10^{-3}$ | | |
| (STABLE BUT INACCURATE) | | | |

obtained. For h = 0.8 to h = 0.9 absolute stability occurs. For h = 1.0 to h = 1.2 oscillation occurred and the solution is inaccurate though stable. For h = 1.3 the error growth is increasing showing actual instability. Thus the method is stable within the range $-1.3 < hC < 0$, which is in conformity with the predicted stability of the general Adams-method, $-1.3 < hC < 0$. This also justifies why most Adams-Moulton methods start with the use of the third order predictor.

8. Fourth Order Adams (P-C-VIII)

Starting value was h = 0.5 with 0.1 increment. Table 9 shows the actual data obtained. From h = 0.5 to h = 0.7 the method is absolutely stable. For h = 0.9 oscillation continues as with h = 0.8. This behavior was observed up to h = 1.0 but the numerical solution is still stable though inaccurate. For h = 1.1 actual instability is evident from the tendency to increasing oscillation of the error. Thus this method is stable for $-1.1 < hC < 0$.

Summarizing then, the real negative stability bounds of the different P-C sets considered are outlined below with $C = f_y(x,y)$ and h = step size.

| P-C Set | Real Negative Stability Limit |
|---|---|
| P-C-I | $-1.9 < hC < 0$ |
| P-C-II | Unstable |
| P-C-III | $-1.8 < hC < 0$ |
| P-C-IV | Unstable |
| P-C-V | $-0.9 < hC < 0$ |
| P-C-VI | $-0.4 < hC < 0$ |
| P-C-VII | $-1.3 < hC < 0$ |
| P-C-VIII | $-1.1 < hC < 0$ |

TABLE 9

ABSOLUTE ERROR (Σ) FOR FOURTH ORDER ADAMS (P-C-VIII)

ODE: $y'=-y$ with $y(0)=1$

Single Precision

| h = 0.5 | | h = 0.7 | |
|---|---|---|---|
| x | $\Sigma_{A4}$ | x | $\Sigma_{A4}$ |
| 2.0 | $5.67 \cdot 10^{-5}$ | 2.8 | $4.094 \cdot 10^{-4}$ |
| 3.0 | $2.125 \cdot 10^{-4}$ | 4.2 | $8.157 \cdot 10^{-4}$ |
| 4.0 | $1.488 \cdot 10^{-4}$ | 5.6 | $3.522 \cdot 10^{-4}$ |
| 5.0 | $8.11 \cdot 10^{-5}$ | 7.0 | $1.239 \cdot 10^{-4}$ |
| 6.0 | $3.95 \cdot 10^{-5}$ | 8.4 | $3.52 \cdot 10^{-5}$ |
| 7.0 | $1.8 \cdot 10^{-5}$ | 9.8 | $7.7 \cdot 10^{-6}$ |
| 8.0 | $7.9 \cdot 10^{-6}$ | (STABLE) | |
| 9.0 | $3.4 \cdot 10^{-6}$ | | |
| 10.0 | $1.4 \cdot 10^{-6}$ | | |
| (STABLE) | | | |

| h = 0.9 | | h = 1.1 | |
|---|---|---|---|
| x | $\Sigma_{A4}$ | x | $\Sigma_{A5}$ |
| 3.6 | $1.819 \cdot 10^{-3}$ | 3.3 | $-3.885 \cdot 10^{-3}$ |
| 5.4 | $2.111 \cdot 10^{-3}$ | 4.4 | $5.789 \cdot 10^{-3}$ |
| 7.2 | $4.93 \cdot 10^{-5}$ | 5.5 | $1.214 \cdot 10^{-3}$ |
| 9.0 | $-5.125 \cdot 10^{-4}$ | 6.6 | $3.818 \cdot 10^{-3}$ |
| (STABLE BUT INACCURATE) | | 7.7 | $6.321 \cdot 10^{-3}$ |
| | | 8.8 | $-4.534 \cdot 10^{-3}$ |
| | | 9.9 | $7.846 \cdot 10^{-3}$ |
| | | (UNSTABLE) | |

The significant points noted from these results are:

i) Within the limits of their stability, the Hamming and Fourth Order Adams methods produced the best accuracy.

ii) It is quite interesting to see that although the Euler and Nystrom methods have the widest range of real negative stability, their results are not quite as accurae as the results of the Hamming and the Fourth Order Adams, even the Third Order Adams methods.

D. NUMERICAL RESULTS FOR RELATIVE STABILITY

If $C > 0$, the solution itself and generally also the error are increasing exponentially. In this situation relative stability is the important consideration. A P-C method is relatively stable if the rate of change of the error with respect to a finite range of integration points is less than the rate of change of the true solution with respect to the same finite range of integration. By this definition it could be seen that it is extremely difficult to establish a fixed bound for relative stability. In order to have a working knowledge of the relative stability of the P-C sets considered the ODE

$$y' = y \quad \text{with} \quad y(0) = 1$$

where $C = f_y(x,y) > 0$ was used. Each P-C set was run with the same series of h values from $h = 0.1$ to $h = 2.0$, with an increment of 0.1 The range of integration for each step size h was $x = 0$ to $x = 10$. Table 10 shows the true solution values from $x = 1$ to $x = 10.0$. Table 10-A to

TABLE 10

TRUE SOLUTION VALUES FOR

ODE: y'=y with y(0)=1
Single Precision

| x | $y_{exact}$ |
|---|---|
| 1.0 | 2.718 |
| 2.0 | 7.389 |
| 3.0 | 20.085 |
| 4.0 | 54.597 |
| 5.0 | 148.409 |
| 6.0 | 403.417 |
| 7.0 | 1096.595 |
| 8.0 | 2980.838 |
| 9.0 | 8102.710 |
| 10.0 | 22025.320 |

Table 10-H show the selected actual data obtained for the different P-C sets.

Table 10-A shows that the Euler method is relatively stable for the particular range of integration. Comparing the error growth with the solution growth of Table 10, it could be seen that the rate of change of error is less than the rate of change of the solution. From h = 0.1 to h = 0.5 the method is quite accurate. However from h = 1.0 to h = 2.0 it becomes quite inaccurate though still relatively stable. Table 10-B shows that the Milne method is not only relatively stable but accurate from h = 0.1 to h = 2.0. From Table 10-C the Nystrom method exhibits relative stability from h = 0.1 to h = 1.0 but becomes unstable at h = 2.0. This could be determined by simply comparing the magnitude of the error at x = 10.0 and h = 2.0 with the magnitude of the true solution at the same point, in which case it is obvious that the error is larger than the true solution. This is another way of looking for relative instability since it is clear that if the rate of change of the error is greater than the rate of change of the solution with respect to n (the number of solution points) then eventually the error will be greater than the true solution. Table 10-D shows that the Hermite method is relatively stable and also accurate. From Table 10-E it can be seen that Hamming's method is also stable and accurate. Table 10-F shows that the second order Adams' method is stable from h = 0.1 to h = 1.0, but the accuracy is quite diminished at h = 1.0. At h = 2.0 it is unstable. From Table 10-G, the third

TABLE 10-A

ABSOLUTE ERROR ($\Sigma$) FOR EULER (P-C-I)

ODE: y'=y with y(0)=1

Single Precision

| x | $\Sigma_E$ | | | |
|---|---|---|---|---|
| | h = 0.1 | h = 0.5 | h = 1.0 | h = 2.0 |
| 1.0 | $-2.231\cdot10^{-3}$ | $-4.819\cdot10^{-2}$ | $-1.567\cdot10^{-1}$ | - |
| 2.0 | $-1.219\cdot10^{-2}$ | $-2.570\cdot10^{-2}$ | $-7.515\cdot10^{-1}$ | -1.610 |
| 3.0 | $-4.980\cdot10^{-2}$ | -1.046 | -2.959 | - |
| 4.0 | $-1.806\cdot10^{-1}$ | -3.808 | -10.638 | -16.401 |
| 5.0 | $-6.142\cdot10^{-1}$ | -13.011 | -36.261 | - |
| 6.0 | -2.004 | -42.721 | -119.355 | -155.571 |
| 7.0 | -6.360 | -136.450 | -383.286 | - |
| 8.0 | -19.774 | -427.070 | -1208.460 | -1420.042 |
| 9.0 | -60.496 | -1316.117 | -3756.500 | - |
| 10.0 | -182.828 | -4006.574 | -11546.150 | -12622.530 |

TABLE 10-B

ABSOLUTE ERROR ($\Sigma$) FOR MILNE (P-C-II)

ODE: y'=y with y(0)=1

Single Precision

| x | $\Sigma_M$ | | | |
|---|---|---|---|---|
| | h = 0.1 | h = 0.5 | h = 1.0 | h = 2.0 |
| 1.0 | $4.8\cdot10^{-6}$ | - | - | - |
| 2.0 | $-2.38\cdot10^{-5}$ | $1.409\cdot10^{-3}$ | - | - |
| 3.0 | $-1.678\cdot10^{-4}$ | $-1.831\cdot10^{-4}$ | - | - |
| 4.0 | $-7.477\cdot10^{-4}$ | $-1.309\cdot10^{-2}$ | $3.070\cdot10^{-1}$ | - |
| 5.0 | $-2.777\cdot10^{-3}$ | $-7.122\cdot10^{-2}$ | $8.342\cdot10^{-1}$ | - |
| 6.0 | $-9.765\cdot10^{-3}$ | $-2.917\cdot10^{-1}$ | 1.794 | - |
| 7.0 | $-3.222\cdot10^{-2}$ | -1.059 | 4.026 | - |
| 8.0 | $-1.040\cdot10^{-1}$ | -3.608 | 8.289 | 362.921 |
| 9.0 | $-3.281\cdot10^{-1}$ | -11.781 | 15.589 | - |
| 10.0 | -1.003 | -37.402 | 23.242 | 3472.707 |

TABLE 10-C

ABSOLUTE ERROR ($\Sigma$) FOR NYSTROM (P-C-III)

ODE: $y'=y$ with $y(0)=1$

Single Precision

| x | $\Sigma_N$ | | | |
|---|---|---|---|---|
| | h = 0.1 | h = 0.5 | h = 1.0 | h = 2.0 |
| 1.0 | $-2.034\cdot10^{-3}$ | $-2.756\cdot10^{-2}$ | - | - |
| 2.0 | $-1.173\cdot10^{-2}$ | $-2.231\cdot10^{-1}$ | $-5.244\cdot10^{-1}$ | - |
| 3.0 | $-4.875\cdot10^{-2}$ | -1.016 | -2.758 | - |
| 4.0 | $-1.783\cdot10^{-1}$ | -3.901 | -11.304 | -16.401 |
| 5.0 | $-6.096\cdot10^{-1}$ | -13.757 | -41.718 | - |
| 6.0 | -1.996 | -46.134 | -145.106 | -215.571 |
| 7.0 | -6.350 | -149.630 | -485.911 | - |
| 8.0 | -19.778 | -473.892 | -1584.737 | -2384.042 |
| 9.0 | -60.589 | -1474.339 | -5069.113 | - |
| 10.0 | -183.289 | -4523.753 | -15975.800 | -24472.530 |

TABLE 10-D

ABSOLUTE ERROR (Σ) FOR HERMITE (P-C-IV)

ODE: y'=y with y(0)=1

Single Precision

| x | $\Sigma_{HE}$ | | | |
|---|---|---|---|---|
| | h = 0.1 | h = 0.5 | h = 1.0 | h = 2.0 |
| 1.0 | $3.8 \cdot 10^{-6}$ | $-3.948 \cdot 10^{-4}$ | - | - |
| 2.0 | $-2.77 \cdot 10^{-5}$ | $-2.104 \cdot 10^{-3}$ | $-1.217 \cdot 10^{-2}$ | - |
| 3.0 | $-1.831 \cdot 10^{-4}$ | $-9.292 \cdot 10^{-3}$ | $-2.285 \cdot 10^{-2}$ | - |
| 4.0 | $-7.629 \cdot 10^{-4}$ | $-3.546 \cdot 10^{-2}$ | $-8.180 \cdot 10^{-2}$ | $7.092 \cdot 10^{-1}$ |
| 5.0 | $-2.853 \cdot 10^{-3}$ | $-1.244 \cdot 10^{-1}$ | $-2.519 \cdot 10^{-1}$ | - |
| 6.0 | $-9.765 \cdot 10^{-3}$ | $-4.147 \cdot 10^{-1}$ | $-7.766 \cdot 10^{-1}$ | 21.334 |
| 7.0 | $-3.247 \cdot 10^{-2}$ | -1.335 | -2.355 | - |
| 8.0 | $-1.044 \cdot 10^{-1}$ | -4.194 | -7.-67 | 275.2517 |
| 9.0 | $-3.281 \cdot 10^{-1}$ | -12.941 | -21.023 | - |
| 10.0 | -1.003 | -39.367 | -62.078 | 2860.671 |

TABLE 10-E

ABSOLUTE ERROR (Σ) FOR HAMMING (P-C-V)

ODE: $y'=y$ with $y(0)=1$

Single Precision

| x | $\Sigma_{HA}$ | | | |
|---|---|---|---|---|
| | h = 0.1 | h = 0.5 | h = 1.0 | h = 2.0 |
| 1.0 | $2.19\cdot10^{-5}$ | - | - | - |
| 2.0 | $6.77\cdot10^{-5}$ | $2.980\cdot10^{-3}$ | - | - |
| 3.0 | $1.831\cdot10^{-5}$ | $2.792\cdot10^{-3}$ | - | - |
| 4.0 | $6.714\cdot10^{-4}$ | $-7.202\cdot10^{-3}$ | $4.332\cdot10^{-1}$ | - |
| 5.0 | $1.831\cdot10^{-3}$ | $-6.004\cdot10^{-2}$ | $8.498\cdot10^{-1}$ | - |
| 6.0 | $5.371\cdot10^{-3}$ | $-2.777\cdot10^{-1}$ | 1.043 | - |
| 7.0 | $1.879\cdot10^{-2}$ | -1.039 | $-5.24\cdot10^{-1}$ | - |
| 8.0 | $4.687\cdot10^{-2}$ | -3.639 | -10.532 | 380.836 |
| 9.0 | $1.406\cdot10^{-1}$ | -12.097 | -53.558 | - |
| 10.0 | $4.296\cdot10^{-1}$ | -38.882 | -213.566 | 3040.449 |

TABLE 10-F

ABSOLUTE ERROR (Σ) FOR SECOND ORDER ADAMS (P-C-VI)

ODE: y'=y with y(0)=1

Single Precision

| x | $\Sigma_E$ | | | |
|---|---|---|---|---|
| | h = 0.1 | h = 0.5 | h = 1.0 | h = 2.0 |
| 1.0 | $-2.100 \cdot 10^{-3}$ | $-2.735 \cdot 10^{-2}$ | - | - |
| 2.0 | $-1.242 \cdot 10^{-2}$ | $-2.358 \cdot 10^{-1}$ | $-5.041 \cdot 10^{-1}$ | - |
| 3.0 | $-5.241 \cdot 10^{-2}$ | -1.127 | -2.792 | - |
| 4.0 | $-1.934 \cdot 10^{-1}$ | -4.453 | -11.936 | -14.401 |
| 5.0 | $-6.649 \cdot 10^{-1}$ | -16.013 | -45.345 | - |
| 6.0 | -2.186 | -54.444 | -161.08 | -201.571 |
| 7.0 | -6.974 | -178.434 | -548.354 | - |
| 8.0 | -21.766 | -569.834 | -1812.745 | -2324.042 |
| 9.0 | -66.800 | -1785.156 | -5866.652 | - |
| 10.0 | -202.378 | -5510.332 | -18684.180 | -24506.530 |

TABLE 10-G

ABSOLUTE ERROR (Σ) FOR THIRD ORDER ADAMS (P-C-VII)

ODE: y'=y with y(0)=1

Single Precision

| x | $\Sigma_{A3}$ | | | |
|---|---|---|---|---|
| | h = 0.1 | h = 0.5 | h - 1.0 | h = 2.0 |
| 1.0 | $2.155 \cdot 10^{-4}$ | - | - | - |
| 2.0 | $1.318 \cdot 10^{-3}$ | $2.076 \cdot 10^{-2}$ | - | - |
| 3.0 | $5.569 \cdot 10^{-3}$ | $1.454 \cdot 10^{-1}$ | $2.131 \cdot 10^{-2}$ | - |
| 4.0 | $2.062 \cdot 10^{-2}$ | $6.352 \cdot 10^{-1}$ | $5.538 \cdot 10^{-1}$ | - |
| 5.0 | $7.075 \cdot 10^{-2}$ | 2.376 | 2.827 | - |
| 6.0 | $2.324 \cdot 10^{-1}$ | 8.216 | 11.147 | 14.938 |
| 7.0 | $7.412 \cdot 10^{-1}$ | 27.090 | 39.645 | - |
| 8.0 | 2.307 | 86.509 | 132.954 | 217.016 |
| 9.0 | 7.089 | 269.992 | 429.273 | - |
| 10.0 | 21.468 | 828.183 | 1349.750 | 2409.128 |

TABLE 10-H

ABSOLUTE ERROR (Σ) FOR FOURTH ORDER ADAMS (P-C-VIII)

ODE: y'=y with y(0)=1

Single Precision

| x | $\Sigma_{A4}$ | | | |
|---|---|---|---|---|
| | h = 0.1 | h = 0.5 | h = 1.0 | h = 2.0 |
| 1.0 | $8.6 \cdot 10^{-6}$ | - | - | - |
| 2.0 | $-2.77 \cdot 10^{-5}$ | $5.53 \cdot 10^{-5}$ | - | - |
| 3.0 | $-1.984 \cdot 10^{-4}$ | $-1.995 \cdot 10^{-2}$ | - | - |
| 4.0 | $-8.087 \cdot 10^{-4}$ | $-1.089 \cdot 10^{-1}$ | $9.992 \cdot 10^{-2}$ | - |
| 5.0 | $-3.234 \cdot 10^{-3}$ | $-4.450 \cdot 10^{-1}$ | $-7.852 \cdot 10^{-1}$ | - |
| 6.0 | $-1.074 \cdot 10^{-2}$ | -1.614 | -5.094 | - |
| 7.0 | $-3.540 \cdot 10^{-2}$ | -5.490 | -21.956 | - |
| 8.0 | $-1.176 \cdot 10^{-1}$ | -17.926 | -81.859 | 262.678 |
| 9.0 | $-3.476 \cdot 10^{-1}$ | -56.878 | -283.230 | - |
| 10.0 | -1.085 | -176.812 | -936.164 | 1748.47 |

order Adams' method showed accuracy and stability from $h = 0.1$ to $h = 2.0$. Finally, Table 10-H shows that the Fourth order Adams' method is quite stable and accurate from $h = 0.1$ to $h = 2.0$.

The most interesting points observed from these actual results are:

i) Though the Milne and Hermite methods have no real negative stability limits, they are quite accurate for $C > 0$. In fact the Hermite method produced the least starting error, and the Milne's method provided the second least starting error, and both maintained accuracy up to $h = 2.0$.

ii) In contrast, the Euler and Nystrom methods both have the widest range of real negative stability limits but showed inaccurate results starting at $h = 1.0$. In fact the Nystrom method is unstable at $h = 2.0$.

iii) The third and fourth order Adams' and Hamming methods showed wide range of relative stability and provided accurate results. The second order Adams' method again showed inaccurate results at $h = 1.0$ and relative instability at $h = 2.0$.

iv) From $h = 0.1$ to $h = 1.0$, the methods in terms of accuracy, rank as follows: Milne ranks first, Hermite second, then Hamming, Fourth Order Adams, and Third Order Adams in that order.

# IX. NUMERICAL EXPERIMENTS

In order to test the performance of the different P-C sets considered, a collection of test ODEs with many unusual and interesting features (i.e., singularities, discontinuities, infinite derivatives, oscillating derivatives, etc.) were selected. Several of such ODEs were presented by Hull and Creemer [Ref. 8] and Lapidus and Seinfeld [Ref. 2]. Table 11 lists the test ODEs considered. To simplify notation, the ODE number will be used to specify the differential equation. For example ODE I is equivalent to the ODE $y' = -y + 10\sin 3x$ with $y(0) = -3$, whose analytic solution is $y(x) = \sin 3x - 3\cos 3x$.

From the previous section on stability bounds of P-C sets it was observed that the different methods showed good accuracy up to $h = 0.5$ on both experimental ODEs, $y' = -y$ and $y' = y$, considered. It was further observed from previous numerical results that all the P-C sets exhibited good stability behavior within the range of h up to $h = 0.5$ except for the Milne and Hermite methods, in the case of real negative stability. But these shortcomings of the Milne and Hermite methods were compensated by the fact that they produced good accuracy and wide range of relative stability. This analysis is needed in the sense that the test ODEs considered exhibit both cases of $f_y(x,y) < 0$ and $f_y(x,y) > 0$. By using values of $h = a^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}$

where the largest value of h = 0.5, each method will be expected to perform quite well. Thus a comparative analysis of their accuracy and computing time would be meaningful. In dealing with accuracy, the effects of the propagation of errors would be strongly felt, as has already been shown in the previous analysis of error propagation. While computing time takes into its fold the number and complexity of function evaluations.

All the eight P-C sets were run on each test ODE using the series of h values as previously mentioned. The Fortran programs used were the same as those used for stability analysis, but the precision used was Double Precision (14 digit precision for IBM 360/67 computer) to minimize the effect of rounding error as h decreases. Each test ODE will be studied individually, with the objective of providing useful and important recommendations on which P-C set performs best for the particular class of problems.

## A. ODE I

Table 12 shows the exact solution from x = 1.0 to x = 10.0. From Table 12-A the results show that the Euler method provides the best accuracy with 7.65 secs in computing time with the Nystrom's ranking second in accuracy though with less computing time, 7.46 secs. The Milne and the Hermite methods are not considered since they both showed instability in their numerical solution values. This could easily be seen by looking at Table 12-B which shows the behavior of the errors for these methods. The errors for

TABLE 11

TEST ODEs

| ODE Number | ODE | Initial Condition | Analytical Solution |
|---|---|---|---|
| I | $y'=-y + 10\sin 3x$ | $y(0) = -3$ | $y(x)=\sin 3x - 3\cos 3x$ |
| II | $y,=y + 2\sin x$ | $y(0) = -1$ | $y(x)=-\sin x - \cos x$ |
| III | $y'=(y-\sin x)\ln(1+ x/40)+ \cos x$ | $y(0) = 0$ | $y(x) = \sin x$ |
| IV | $y'=-xy/(4x+16)$ | $y(0) = 4$ | $y(x) = (x+4)e^{-x/4}$ |
| V | $y'=(1+y^2)/2(2500-x^2)^{\frac{1}{2}}$ | $y(0) = 1$ | $y(x)=((50+x)/(50-x))^{\frac{1}{2}}$ |
| VI | $y'=y - 2x/y$ | $y(0) = 1$ | $y(x) = (2x+1)^{\frac{1}{2}}$ |
| VII | $y'=1 - y^2$ | $y(0) = 0$ | $y(x)=(e^{2x}-1)/(e^{2x}+1)$ |
| VIII | $y'=y\cos x$ | $y(0) = 1$ | $y(x) = e^{\sin x}$ |
| IX | $y'=y\cos^2 x$ | $y(0) = 1$ | $y(x)=e^{(x/2 + 1/4 \sin 2x)}$ |
| X | $y'=-1.38y - 0.81(.001 e^{-3x}+ 4e^{-.3x})$ | $y(0)=-2.9995$ | $y(x)=0.0005 e^{-3x}- 3e^{-.3x}$ |

TABLE 12

TRUE SOLUTION VALUES FOR ODE I
Double Precision

| x | $y_{exact}$ |
|---|---|
| 1.0 | 3.111097 |
| 2.0 | -3.159926 |
| 3.0 | 3.145509 |
| 4.0 | -3.068134 |
| 5.0 | 2.929351 |
| 6.0 | -2.731937 |
| 7.0 | 2.479843 |
| 8.0 | -2.178115 |
| 9.0 | 1.832792 |
| 10.0 | -1.450785 |

TABLE 12-A

ODE I AT x = 10.0

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-III Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | $-9.83479\cdot10^{-5}$ | $4.51483\cdot10^{-9}$ | $-9.83522\cdot10^{-5}$ | $-8.53288\cdot10^{-3}$ |
| 64 | $-3.93360\cdot10^{-4}$ | $8.21731\cdot10^{-8}$ | $-3.93415\cdot10^{-4}$ | $-3.40492\cdot10^{-2}$ |
| 32 | $-1.57286\cdot10^{-3}$ | $-1.17807\cdot10^{-7}$ | $-1.57372\cdot10^{-3}$ | $-1.35592\cdot10^{-1}$ |
| 16 | $-6.28126\cdot10^{-3}$ | $-2.51544\cdot10^{-5}$ | $-6.92664\cdot10^{-3}$ | $-5.38364\cdot10^{-1}$ |
| 8 | $-2.49436\cdot10^{-2}$ | $-1.35324\cdot10^{-3}$ | $-2.52237\cdot10^{-2}$ | -2.134035 |
| 4 | $-9.64382\cdot10^{-2}$ | $-3.35324\cdot10^{-2}$ | $-1.00786\cdot10^{-1}$ | -6.039341 |
| 2 | $-3.34570\cdot10^{-1}$ | 1.527752 | $-3.38212\cdot10^{-1}$ | -37.089294 |
| CPTS* | 7.65 | 5.59 | 7.46 | 8.37 |

*CPTS means computing time in seconds.

TABLE 12-B

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV

Double Precision

ODE I; h = 0.5

| x | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 1.0 | $7.171 \cdot 10^{-1}$ | - | $4.159 \cdot 10^{-1}$ | -1.584 |
| 2.0 | $-4.640 \cdot 10^{-1}$ | $2.081 \cdot 10^{-1}$ | $-6.649 \cdot 10^{-1}$ | -1.645 |
| 3.0 | $5.756 \cdot 10^{-1}$ | $9.447 \cdot 10^{-2}$ | $5.817 \cdot 10^{-1}$ | -2.464 |
| 4.0 | $-5.392 \cdot 10^{-1}$ | $2.509 \cdot 10^{-1}$ | $-6.091 \cdot 10^{-1}$ | -3.386 |
| 5.0 | $5.418 \cdot 10^{-1}$ | $2.076 \cdot 10^{-1}$ | $5.806 \cdot 10^{-1}$ | -5.218 |
| 6.0 | $-5.154 \cdot 10^{-1}$ | $4.266 \cdot 10^{-1}$ | $-5.560 \cdot 10^{-1}$ | -7.537 |
| 7.0 | $4.854 \cdot 10^{-1}$ | $4.737 \cdot 10^{-1}$ | $5.148 \cdot 10^{-1}$ | -11.400 |
| 8.0 | $-4.431 \cdot 10^{-1}$ | $7.910 \cdot 10^{-1}$ | $-4.652 \cdot 10^{-1}$ | -11.747 |
| 9.0 | $3.929 \cdot 10^{-1}$ | 1.011 | $4.056 \cdot 10^{-1}$ | -25.066 |
| 10.0 | -3.345 | 1.527 | $-3.382 \cdot 10^{-1}$ | -37.089 |

the Milne and Hermite methods increased in one direction eventually growing to magnitudes greater than the true solution at x = 10.0 (i.e., for Milne's error, 1.527 > 1.450 and for Hermite's error -37.089 > 1.450), thus these methods are not suited for solving ODE I. The Euler and Nystrom methods both started at accuracy approximately equal to fifth decimal places and ended up at h = 0.5 with accuracy up to the first decimal place. But though their accuracy is quite restrained as h increases, nevertheless the behavior of the error tends to decrease as the range of integration is increased. From this group then the Euler method is the best choice for ODE I.

In Table 12-C the results for the other four methods are listed. All these methods showed stability for all values of h used, but Hamming's produced the best accuracy and the best computing time. Hamming's method started at $10^{-10}$ accuracy at 1/h = 128 and ended with $10^{-2}$ at h = 0.5. The Fourth Order Adams' comes next in both accuracy and computing time, then the Third Order Adams', and the Second Order Adams' came in that order.

Comparing these last four methods with the Euler method, only Adams' second order method is inferior. Therefore, for solving ODEs that belong to the class of ODE I the Hamming method is highly recommended in terms of accuracy and least cost in computer time. The Fourth Order Adams' is the next choice in the absence of the Hamming method.

TABLE 12-C

ODE I at x = 10.0

ABSOLUTE ERROR (Σ) FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII

Double Precision

| 1/h | $\varepsilon_{Ha}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | $3.0381 \cdot 10^{-10}$ | $-9.86551 \cdot 10^{-5}$ | $-9.60790 \cdot 10^{-7}$ | $1.86070 \cdot 10^{-8}$ |
| 64 | $1.12626 \cdot 10^{-8}$ | $-3.95790 \cdot 10^{-4}$ | $-7.81167 \cdot 10^{-6}$ | $3.18596 \cdot 10^{-7}$ |
| 32 | $4.68494 \cdot 10^{-7}$ | $-1.59200 \cdot 10^{-3}$ | $-6.45309 \cdot 10^{-5}$ | $4.04497 \cdot 10^{-6}$ |
| 16 | $1.01544 \cdot 10^{-5}$ | $-6.42959 \cdot 10^{-3}$ | $-5.49889 \cdot 10^{-4}$ | $5.91417 \cdot 10^{-5}$ |
| 8 | $3.69762 \cdot 10^{-4}$ | $-2.60335 \cdot 10^{-2}$ | $-4.97320 \cdot 10^{-3}$ | $7.48403 \cdot 10^{-4}$ |
| 4 | $1.29180 \cdot 10^{-2}$ | $-1.03170 \cdot 10^{-1}$ | $-5.00046 \cdot 10^{-2}$ | $4.56055 \cdot 10^{-3}$ |
| 2 | $6.39906 \cdot 10^{-2}$ | -3.586413 | $-4.74203 \cdot 10^{-1}$ | $-1.90291 \cdot 10^{-1}$ |
| CPTS | 5.76 | 7.98 | 8.55 | 5.99 |

## B. ODE II

The true solution values are given in Table 13 from $x = 1.0$ to $x = 10.0$. From the data presented in Table 13-A it is clear that the most important consideration is the step size h. For h = 0.5 the four methods identically yield errors greater than the true solution at the same point. In solving problems of this type then h must be chosen to be smaller than 0.25 for the numerical solution to be valid using the Euler, Milne, and Nystrom methods. The Hermite method obviously is not worth considering, because of its inaccurate results for all values of h. For $h < 0.25$ the Milne method gives the best numerical solution and the least computing time, followed by the Euler then the Nystrom methods.

Table 13-B shows that to have a vlid numerical solution for any of the other four methods h must be less than 0.25. For values of $h < 0.25$ , the Hamming method yields the greatest accuracy, followed by the Fourth, Second, and Third Order Adams'. The Milne method is a little better than the Second Order Adams'. Thus for accuracy the Hamming method ranks first, then the Fourth Order Adams, Milne, Euler, Second Order Adams, Third Order Adams, and lastly the Nystrom methods.

For problems in the class of ODE it is recommended that h must be chosen less than 0.25 or smaller if valid numerical solution and accuracy are desired. Then use Hamming as the numerical method to solve the ODE. Again the Fourth Order Adams method is the second best choice. The choice of h can

TABLE 13

TRUE SOLUTION VALUES FOR ODE II
Double Precision

| x | $y_{exact}$ |
|---|---|
| 1.0 | -1.381773 |
| 2.0 | $-4.931505 \cdot 10^{-1}$ |
| 3.0 | $8.488724 \cdot 10^{-1}$ |
| 4.0 | 1.410446 |
| 5.0 | $6.752620 \cdot 10^{-1}$ |
| 6.0 | $-6.807547 \cdot 10^{-1}$ |
| 7.0 | -1.410888 |
| 8.0 | $-8.438582 \cdot 10^{-1}$ |
| 9.0 | $4.990117 \cdot 10^{-1}$ |
| 10.0 | 1.383092 |

TABLE 13-A

ODE II at $x = 10.0$

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV
Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | $2.596512 \cdot 10^{-3}$ | $2.50786 \cdot 10^{-9}$ | $8.444203 \cdot 10^{-4}$ | $-4.446340 \cdot 10^{-1}$ |
| 64 | $2.518638 \cdot 10^{-3}$ | $1.634916 \cdot 10^{-7}$ | $7.950519 \cdot 10^{-3}$ | -1.764591 |
| 32 | $2.726949 \cdot 10^{-3}$ | $9.921730 \cdot 10^{-6}$ | $5.455366 \cdot 10^{-2}$ | -6.947361 |
| 16 | $4.095807 \cdot 10^{-2}$ | $5.693167 10^{-4}$ | $4.234564 \cdot 10^{-1}$ | -26.911798 |
| 8 | $6.213302 \cdot 10^{-1}$ | $2.178410 \cdot 10^{-3}$ | 3.214692 | -100.834152 |
| 4 | 9.160642 | $1.362692 \cdot 10^{-1}$ | 23.407199 | -353.009679 |
| 2 | 131.604044 | 5.482274 | 157.762746 | -1081.991963 |
| CPTS | 6.38 | 4.69 | 5.34 | 8.26 |

TABLE 13-B

ODE at x = 10.0

ABSOLUTE ERROR FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII

Double Precision

| 1/h | $\Sigma_{HA}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | $-1.20446 \cdot 10^{-9}$ | $-3.760418 \cdot 10^{-4}$ | $4.448046 \cdot 10^{-4}$ | $-2.334506 \cdot 10^{-8}$ |
| 64 | $-7.48373 \cdot 10^{-9}$ | $-3.192068 \cdot 10^{-3}$ | $3.610822 \cdot 10^{-3}$ | $-6.282905 \cdot 10^{-7}$ |
| 32 | $5.531148 \cdot 10^{-7}$ | $-2.331998 \cdot 10^{-2}$ | $2.961989 \cdot 10^{-2}$ | $-1.347495 \cdot 10^{-5}$ |
| 16 | $7.353901 \cdot 10^{-5}$ | $-1.532622 \cdot 10^{-1}$ | $2.455737 \cdot 10^{-1}$ | $-6.597818 \cdot 10^{-5}$ |
| 8 | $3.686296 \cdot 10^{-3}$ | $-7.673571 \cdot 10^{-1}$ | 2.016912 | $-1.600424 \cdot 10^{-2}$ |
| 4 | $1.657253 \cdot 10^{-1}$ | $-6.706531 \cdot 10^{-1}$ | 15.053093 | $-2.268373 \cdot 10^{-1}$ |
| 2 | 6.292332 | 44.871662 | 80.187459 | 1.401846 |
| CPTS | 5.26 | 7.79 | 8.36 | 4.96 |

be formally stated as $|hC| < 0.25$, since the value of h to be used is directly dependent upon $C = f_y(x,y)$.

C. ODE III

Table 14 shows the exact solution values for ODE III from x = 1.0 to x = 10.0. From the data of Table 14-A it is clear that all four methods performed quite well to a good degree of accuracy for all balues of h. But for best accuracy and least computing time the Milne Method stands out followed by the Euler, Hermite, and Nystrom methods in that order.

From Table 14-B, again it is observed that any of the last four methods yield a valid numerical solution, to a certain degree of accuracy. For an excellent accuracy, however, the Hamming method is the best choice. Its range of accuracy is from $10^{-12}$ up to $10^{-3}$ for values of h ranging from 1/h = 128 to 1/h = 2. The Milne method compared to these last four methods ranks second only to the Hamming, though its computing time is a little smaller than Hamming's. Thus for excellent accuracy, the order of choice is an follows: Hamming, Milne, Fourth Order Adams, Third Order Adams, Hermite, Euler, Second Order Adams, and lastly the Nystrom method in solving ODEs belonging to the class of ODE III.

D. ODE IV

True solution values are listed in Table 15 from x = 1.0 to x = 10.0. Actual data obtained in Table 15-A showed that all four P-C sets are good numerical methods for ODE IV. But if a choice is to be made the Milne's accuracy is far greater than the other three methods with computing time only 0.16

TABLE 14

TRUE SOLUTION VALUES FOR ODE III

Double Precision

| $x$ | $y_{exact}$ |
|---|---|
| 1.0 | $8.414709 \cdot 10^{-1}$ |
| 2.0 | $9.092974 \cdot 10^{-1}$ |
| 3.0 | $1.411200 \cdot 10^{-1}$ |
| 4.0 | $-7.568024 \cdot 10^{-1}$ |
| 5.0 | $-9.589242 \cdot 10^{-1}$ |
| 6.0 | $-2.794154 \cdot 10^{-1}$ |
| 7.0 | $6.569865 \cdot 10^{-1}$ |
| 8.0 | $9.893582 \cdot 10^{-1}$ |
| 9.0 | $4.121184 \cdot 10^{-1}$ |
| 10.0 | $-5.440211 \cdot 10^{-1}$ |

TABLE 14-A

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV

ODE III AT x=10.0

Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | $-1.705737\cdot10^{-6}$ | $-1.099\cdot10^{-11}$ | $-1.785639\cdot10^{-6}$ | $1.341400\cdot10^{-7}$ |
| 64 | $-6.845247\cdot10^{-6}$ | $9.286\cdot10^{-11}$ | $-7.814645\cdot10^{-6}$ | $1.288789\cdot10^{-6}$ |
| 32 | $-2.703738\cdot10^{-5}$ | $1.26041\cdot10^{-9}$ | $-3.531438\cdot10^{-5}$ | $1.031052\cdot10^{-5}$ |
| 16 | $-1.089843\cdot10^{-4}$ | $1.58770\cdot10^{-8}$ | $-1.736454\cdot10^{-4}$ | $8.247845\cdot10^{-5}$ |
| 8 | $-4.365210\cdot10^{-4}$ | $1.596440\cdot10^{-6}$ | $-9.528314\cdot10^{-4}$ | $6.594418\cdot10^{-4}$ |
| 4 | $-1.755574\cdot10^{-3}$ | $4.317133\cdot10^{-5}$ | $-5.856889\cdot10^{-3}$ | $5.260049\cdot10^{-3}$ |
| 2 | $-7.174787\cdot10^{-3}$ | $1.071599\cdot10^{-3}$ | $-3.910558\cdot10^{-2}$ | $4.154229\cdot10^{-2}$ |
| CPTS | 7.81 | 5.59 | 7.12 | 7.92 |

TABLE 14-B

ODE AT x = 10.0

ABSOLUTE ERROR FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII
Double Precision

| 1/h | $\Sigma_{HA}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | $2.04\cdot10^{-12}$ | $5.023475\cdot10^{-6}$ | $-1.900550\cdot10^{-8}$ | $3.683\cdot10^{-11}$ |
| 64 | $6.396\cdot10^{-11}$ | $-7.798238\cdot10^{-6}$ | $-6.720151\cdot10^{-7}$ | $6.5237\cdot10^{-10}$ |
| 32 | $2.01370\cdot10^{-9}$ | $-3.512798\cdot10^{-5}$ | $-5.373282\cdot10^{-6}$ | $1.246830\cdot10^{-8}$ |
| 16 | $6.236927\cdot10^{-8}$ | $-1.721024\cdot10^{-4}$ | $-4.287630\cdot10^{-5}$ | $2.649671\cdot10^{-7}$ |
| 8 | $1.970409\cdot10^{-6}$ | $-9.408691\cdot10^{-4}$ | $-3.391988\cdot10^{-4}$ | $6.946177\cdot10^{-6}$ |
| 4 | $6.368642\cdot10^{-5}$ | $-5.767408\cdot10^{-3}$ | $-2.588546\cdot10^{-3}$ | $1.805833\cdot10^{-4}$ |
| 2 | $1.569286\cdot10^{-3}$ | $-3.848313\cdot10^{-2}$ | $-1.686135\cdot10^{-2}$ | $4.360227\cdot10^{-3}$ |
| CPTS | 6.23 | 10.0 | 10.0 | 8.23 |

TABLE 15

TRUE SOLUTION VALUES FOR ODE IV
Double Precision

| x | $y_{exact}$ |
|---|---|
| 1.0 | 3.894003 |
| 2.0 | 3.639183 |
| 3.0 | 3.306565 |
| 4.0 | 2.943035 |
| 5.0 | 2.578543 |
| 6.0 | 2.231301 |
| 7.0 | 1.911513 |
| 8.0 | 1.624023 |
| 9.0 | 1.370189 |
| 10.0 | 1.149189 |

TABLE 15-A

ODE IV at x = 10.0

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV

Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | $-7.344900 \cdot 10^{-7}$ | $-3.4 \cdot 10^{-3}$ | $-4.598339 \cdot 10^{-7}$ | $4.575814 \cdot 10^{-6}$ |
| 64 | $-1.839350 \cdot 10^{-6}$ | $-6.07 \cdot 10^{-12}$ | $-1.840047 \cdot 10^{-6}$ | $1.826764 \cdot 10^{-5}$ |
| 32 | $-7.335362 \cdot 10^{-6}$ | $-1.0603 \cdot 10^{-10}$ | $-7.339871 \cdot 10^{-6}$ | $7.278726 \cdot 10^{-5}$ |
| 16 | $-2.926381 \cdot 10^{-5}$ | $-1.97060 \cdot 10^{-9}$ | $-2.872267 \cdot 10^{-5}$ | $2.889080 \cdot 10^{-4}$ |
| 8 | $-1.176812 \cdot 10^{-4}$ | $-4.063982 \cdot 10^{-8}$ | $-1.119506 \cdot 10^{-4}$ | $1.128099 \cdot 10^{-3}$ |
| 4 | $-4.709013 \cdot 10^{-4}$ | $-6.024899 \cdot 10^{-7}$ | $-4.260573 \cdot 10^{-4}$ | $4.418111 \cdot 10^{-3}$ |
| 2 | $-1.885932 \cdot 10^{-3}$ | $-5.791009 \cdot 10^{-6}$ | $-1.541727 \cdot 10^{-3}$ | $1.668370 \cdot 10^{-2}$ |
| CPTS | 4.39 | 3.83 | 3.67 | 6.74 |

sec longer than the next best choice which is Nystrom. After Nystrom, the Euler method is preferred, then the Hermite.

Table 15-B data again demonstrates that any of the last four P-C sets can be used for solving ODE IV. Milne method compared with these methods competes with the Hamming in accuracy and has an edge in computer time (3.83 secs < 4.45 secs). But this fraction of seconds difference is overcome by the demonstrated better accuracy of the Hamming in every step of the solution from $h = 2^{-7}$ to $h = 2^{-1}$. Thus, if preference is to be made, Hamming is the most likely choice as the best method for solving ODE IV class of problems. The Milne, Fourth Order Adams, Third Order Adams, Nystrom, Second Order Adams, Euler, and Hermite are the order of choices following the Hamming method.

E. ODE V

Table 16 shows the true solution values for the range of integration considered. From the data of Tables 16-A and 16-B, it is obvious that any method can be used to solve ODE V and obtained accuracy up to minimum of $10^{-6}$ and maximum of $10^{-14}$. Thus in comparing these methods, accuracy criteria must not be the greatest concern. By studying closely the behavior of the errors as h increases, it was noted that the Milne, Hamming, and Fourth Order Adams exhibited decreasing errors from $h = 2^{-7}$ to $h = 2^{-5}$ for Milne, and from $h = 2^{-7}$ to $h = 2^{-4}$ for both Hamming and Fourth Order Adams. Analyzing further the Hamming and Fourth Order Adams methods it could be seen that the Fourth

TABLE 15-B

ODE IV at x = 10.0

ABSOLUTE ERROR FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII
Double Precision

| 1/h | $\Sigma_{HA}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | $3.0\cdot10^{-13}$ | $-4.585327\cdot10^{-7}$ | $1.28499\cdot10^{-9}$ | $-1.61\cdot10^{-12}$ |
| 64 | $2.4\cdot10^{-13}$ | $-1.829584\cdot10^{-6}$ | $1.025906\cdot10^{-8}$ | $-2.690\cdot10^{-11}$ |
| 32 | $2.37\cdot10^{-12}$ | $-7.282200\cdot10^{-6}$ | $8.174763\cdot10^{-8}$ | $-4.3747\cdot10^{-10}$ |
| 16 | $7.849\cdot10^{-11}$ | $-2.884380\cdot10^{-5}$ | $6.488701\cdot10^{-7}$ | $-7.20867\cdot10^{-9}$ |
| 8 | $2.78944\cdot10^{-9}$ | $-1.131593\cdot10^{-4}$ | $5.111696\cdot10^{-6}$ | $-1.225493\cdot10^{-7}$ |
| 4 | $1.186192\cdot10^{-7}$ | $-4.359162\cdot10^{-4}$ | $3.970077\cdot10^{-5}$ | $1.778276\cdot10^{-6}$ |
| 2 | $2.202890\cdot10^{-6}$ | $-1.625329\cdot10^{-3}$ | $3.006535\cdot10^{-4}$ | $2.201592\cdot10^{-5}$ |
| CPTS | 4.45 | 6.46 | 7.02 | 4.02 |

TABLE 16

TRUE SOLUTION VALUES FOR ODE V

Double Precision

| $x$ | $y_{exact}$ |
|---|---|
| 1.0 | 1.020204 |
| 2.0 | 1.040832 |
| 3.0 | 1.061913 |
| 4.0 | 1.083472 |
| 5.0 | 1.105541 |
| 6.0 | 1.128152 |
| 7.0 | 1.151338 |
| 8.0 | 1.175139 |
| 9.0 | 1.199593 |
| 10.0 | 1.224744 |

TABLE 16-A

ODE V at x = 10.0

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV

Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | $2.26254 \cdot 10^{-9}$ | $7.0 \cdot 10^{-14}$ | $-1.94002 \cdot 10^{-9}$ | $-4.8 \cdot 10^{-13}$ |
| 64 | $9.04072 \cdot 10^{-9}$ | $4.0 \cdot 10^{-14}$ | $-7.747767 \cdot 10^{-8}$ | $-4.45 \cdot 10^{-12}$ |
| 32 | $3.609166 \cdot 10^{-8}$ | 0* | $-3.088771 \cdot 10^{-8}$ | $-3.583 \cdot 10^{-11}$ |
| 16 | $3.436058 \cdot 10^{-8}$ | $-2.2 \cdot 10^{-13}$ | $-1.227293 \cdot 10^{-7}$ | $-2.8689 \cdot 10^{-10}$ |
| 8 | $4.960094 \cdot 10^{-7}$ | $-3.28 \cdot 10^{-12}$ | $-4.844143 \cdot 10^{-7}$ | $-2.29641 \cdot 10^{-9}$ |
| 4 | $1.978142 \cdot 10^{-6}$ | $-4.237 \cdot 10^{-11}$ | $-1.886736 \cdot 10^{-6}$ | $-1.839847 \cdot 10^{-8}$ |
| 2 | $7.866820 \cdot 10^{-6}$ | $-4.0408 \cdot 10^{-10}$ | $-7.644850 \cdot 10^{-6}$ | $-1.478586 \cdot 10^{-7}$ |
| CPTS | 5.63 | 5.58 | 5.16 | 5.35 |

*0 means error less than $10^{-14}$.

TABLE 16-B

ODE V at x = 10.0

ABSOLUTE ERROR FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII

Double Precision

| 1/h | $\Sigma_{HA}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | $4.0 \cdot 10^{-13}$ | $-9.29021 \cdot 10^{-9}$ | $-6.0 \cdot 10^{-14}$ | $1.6 \cdot 10^{-13}$ |
| 64 | $1.9 \cdot 10^{-13}$ | $-1.179555 \cdot 10^{-8}$ | $-2.46 \cdot 10^{-12}$ | $1.1 \cdot 10^{-13}$ |
| 32 | $1.2 \cdot 10^{-13}$ | $-3.331471 \cdot 10^{-8}$ | $-2.018 \cdot 10^{-11}$ | $7.0 \cdot 10^{-14}$ |
| 16 | $5.0 \cdot 10^{-14}$ | $-1.249903 \cdot 10^{-7}$ | $-1.6114 \cdot 10^{-10}$ | $-3.0 \cdot 10^{-14}$ |
| 8 | $-1.4 \cdot 10^{-13}$ | $-4.926236 \cdot 10^{-7}$ | $-1.27940 \cdot 10^{-9}$ | $-1.14 \cdot 10^{-12}$ |
| 4 | $-4.83 \cdot 10^{-12}$ | $-1.950214 \cdot 10^{-6}$ | $-1.007531 \cdot 10^{-8}$ | $-1.802 \cdot 10^{-11}$ |
| 2 | $-1.1744 \cdot 10^{-10}$ | $-7.649028 \cdot 10^{-6}$ | $-7.798880 \cdot 10^{-8}$ | $-2.7389 \cdot 10^{-10}$ |
| CPTS | 6.31 | 10.0 | 10.0 | 7.78 |

Order Adams had its optimum value of-h at $h < 2^{-4}$ while that of Hamming achieved its optimum h value at $h < 2^{-3}$. Thus the Hamming method has a higher value of h for its range of excellent accuracy. This property gives the Hamming method a slight edge over that of Milne, and the Fourth Order Adams. However if both accuracy and computing time are considered the Milne method is likely to be the choice. Therefore for the class of ODE V, the recommendation for the best method to be used will be most likely dependent upon the particular interest: whether accuracy and wider range of h values are the primary concern, or whether accuracy and least computing time is the criterion. For the former criteria the Hamming method serves the best purpose while for the latter the Milne method offers the best solution. However, as was noted earlier, if no criterion is involved but the interest is just to solve the problem the easiest way, the self-starting Euler method is recommended. For realistic purposes however the following order of choice is highly recommended: Hamming, Milne, Fourth Order Adams, Third Order Adams, Hermite, Nystrom, Euler, and lastly Second Order Adams.

F. ODE VI

True solution values are listed in Table 17 from x = 1.0 to x = 10.0. Results from Table 17-A clearly showed that h must be chosen to be quite small to have a valid numerical solution. For the series of h values chosen none of the methods exhibited valid numerical solution from 1/h = 32 up

TABLE 17

TRUE SOLUTION VALUES FOR ODE VI

Double Precision

| x | $y_{exact}$ |
|---|---|
| 1.0 | 1.732050 |
| 2.0 | 2.236067 |
| 3.0 | 2.645751 |
| 4.0 | 3.0 |
| 5.0 | 3.316624 |
| 6.0 | 3.605551 |
| 7.0 | 3.872983 |
| 8.0 | 4.123105 |
| 9.0 | 4.358898 |
| 10.0 | 4.582575 |

TABLE 17-A

ODE VI at $x = 10.0$

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV

Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | -50.844 | $-9.503 \cdot 10^{-3}$ | -47.611 | -32.764 |
| 64 | -107.099 | $7.761 \cdot 10^{-2}$ | -99.681 | -1.885 |
| 32 | -214.685 | 6.986 | -198.306 | 18518.873 |
| 16 | -436.640 | -12.200 | -373.796 | 382.107 |
| 8 | -898.232 | -55.310 | -660.199 | -3060.609 |
| 4 | -1940.153 | -210.374 | -1051.118 | -2279.396 |
| 2 | -4695.585 | -879.472 | -1475.333 | -4096.870 |
| CPTS | 5.65 | 4.40 | 5.12 | 7.11 |

to $h = 2^{-1}$. The Milne method provided a valid numerical solution starting at $h = 2^{-6}$ and smaller, but the other methods still are unreliable.

From Table 17-B, only the Hamming and Fourth Order Adams method showed a valid numerical solution for h $2^{-6}$ and smaller, all others need much smaller h than the lowest value, $2^{-7}$ chosen. Again on the average comparing the Milne, Hamming, and the Fourth Order Adams, for use in solving ODE VI with values of h $2^{-6}$, the Hamming is the first choice, Milne second and the Fourth Order Adams. As in the case of ODE II, the h values are governed by the maximum magnitude of $C = f_y(x,y)$. If C is large then h must be chosen to be very small to satisfy the bounds for stability as had been established before.

G. ODE VII

True solution values are shown in Table 18 from x = 1.0 to x = 10.0. Table 18-A showed that the Euler and Nystrom provide the accurate numerical solutions desired while the Milne and Hermite are unreliable. By analyzing the behavior of the roots of the Milne and Hermite methods shown in Table 18-B, it was observed that both exhibited unstable solutions as evidenced by the uniform growth of the error in one direction while that of the Nystrom and Euler had decreasing error as the range of iteration increases. Between the Euler and Nystrom, the former produced more accurate results though it was a fraction of a second longer in computing time than the Nystrom.

TABLE 17-B

ODE VI at x = 10.0

ABSOLUTE ERROR FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII
Double Precision

| 1/h | $\Sigma_{HA}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | $3.848 \cdot 10^{-3}$ | -48.786 | -16.423 | $-9.371 \cdot 10^{-2}$ |
| 64 | $1.073 \cdot 10^{-1}$ | -99.150 | -38.845 | $-9.692 \cdot 10^{-1}$ |
| 32 | 4.256 | -191.895 | -48.587 | -4.337 |
| 16 | 4.556 | -347.874 | -45.839 | -34.281 |
| 8 | -26.070 | -557.063 | 188.851 | -100.571 |
| 4 | -129.471 | -613.784 | 295.289 | -255.149 |
| 2 | -866.001 | -52.022 | 3621.438 | -897.303 |
| CPTS | 5.0 | 6.68 | 7.18 | 4.79 |

TABLE 18

TRUE SOLUTION VALUES FOR ODE VII

Double Precision

| x | $y_{exact}$ |
|---|---|
| 1.0 | $7.615941 \cdot 10^{-1}$ |
| 2.0 | $9.640275 \cdot 10^{-1}$ |
| 3.0 | $9.950547 \cdot 10^{-1}$ |
| 4.0 | $9.993292 \cdot 10^{-1}$ |
| 5.0 | $9.999092 \cdot 10^{-1}$ |
| 6.0 | $9.999877 \cdot 10^{-1}$ |
| 7.0 | $9.999983 \cdot 10^{-1}$ |
| 8.0 | $9.999997 \cdot 10^{-1}$ |
| 9.0 | $9.999999 \cdot 10^{-1}$ |
| 10.0 | $9.999999 \cdot 10^{-1}$ |

TABLE 18-A

ODE VII at x = 10.0

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV

Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | $5.91 \cdot 10^{-2}$ | $-1.18551 \cdot 10^{-9}$ | $-1.56 \cdot 10^{-12}$ | $-1.750057 \cdot 10^{-5}$ |
| 64 | $2.092 \cdot 10^{-11}$ | $-6.824836 \cdot 10^{-8}$ | $-6.78 \cdot 10^{-12}$ | $-2.691970 \cdot 10^{-5}$ |
| 32 | $7.349 \cdot 10^{-11}$ | $-5.977050 \cdot 10^{-6}$ | $-3.051 \cdot 10^{-11}$ | $-3.963316 \cdot 10^{-5}$ |
| 16 | $2.5577 \cdot 10^{-10}$ | $-6.000472 \cdot 10^{-5}$ | $-1.4540 \cdot 10^{-10}$ | $-1.511858 \cdot 10^{-5}$ |
| 8 | $8.8764 \cdot 10^{-10}$ | $-3.322982 \cdot 10^{-4}$ | $-7.3452 \cdot 10^{-10}$ | $-9.699544 \cdot 10^{-4}$ |
| 4 | $3.58311 \cdot 10^{-9}$ | $3.314797 \cdot 10^{-3}$ | $-3.46305 \cdot 10^{-9}$ | $-6.187207 \cdot 10^{-2}$ |
| 2 | $-1.07753 \cdot 10^{-9}$ | 1.429417 | $-5.593088 \cdot 10^{-7}$ | $3.852296 \cdot 10^{-1}$ |
| CPTS | 4.42 | 4.75 | 4.14 | 6.63 |

TABLE 18-B

ODE at h = 0.5

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, P-C-IV

Double Precision

| x | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 1.0 | $1.153\cdot10^{-2}$ | - | $-4.193\cdot10^{-5}$ | $-2.027\cdot10^{-4}$ |
| 2.0 | $8.578\cdot10^{-3}$ | $1.007\cdot10^{-3}$ | $-1.076\cdot10^{-2}$ | $-6.422\cdot10^{-4}$ |
| 3.0 | $2.967\cdot10^{-3}$ | $-3.266\cdot10^{-3}$ | $-6.486\cdot10^{-3}$ | $-6.869\cdot10^{-4}$ |
| 4.0 | $7.349\cdot10^{-4}$ | $-1.193\cdot10^{-2}$ | $-1.726\cdot10^{-3}$ | $-2.874\cdot10^{-3}$ |
| 5.0 | $1.589\cdot10^{-4}$ | $-2.764\cdot10^{-2}$ | $-2.717\cdot10^{-4}$ | $-8.340\cdot10^{-3}$ |
| 6.0 | $3.208\cdot10^{-5}$ | $-6.232\cdot10^{-2}$ | $-2.363\cdot10^{-5}$ | $-2.261\cdot10^{-2}$ |
| 7.0 | $6.221\cdot10^{-6}$ | $-1.250\cdot10^{-1}$ | $-1.228\cdot10^{-6}$ | $-5.534\cdot10^{-2}$ |
| 8.0 | $9.152\cdot10^{-7}$ | $-1.906\cdot10^{-1}$ | $-1.132\cdot10^{-6}$ | $-1.000\cdot10^{-1}$ |
| 9.0 | $-1.828\cdot10^{-8}$ | $-3.584\cdot10^{-1}$ | $-9.497\cdot10^{-7}$ | $-3.718\cdot10^{-1}$ |
| 10.0 | $-1.077\cdot10^{-9}$ | 1.429 | $-5.593\cdot10^{-7}$ | $3.852\cdot10^{-1}$ |

Results from Table 18-C showed that the last four methods produced valid numerical solutions with Hamming and Fourth Order Adams competing for best accuracy. As h increases from $h = 2^{-7}$ to $h = 2^{-5}$ the Hamming method is most precise but from $h = 2^{-4}$ to $h = 2^{-1}$ the Fourth Order Adams method showed better accuracy. Computing time for the fourth Order Adams method is less than the Hamming method. The Euler method compared with these two methods as h increases, started at lesser accuracy from $h = 2^{-7}$ to $h = 2^{-4}$ but maintained its accurate results up to $h = 2^{-1}$ and yielded a much more accurate numerical solution at $h = 2^{-1}$. It also had the least computing time. Thus for classes of ODE belonging to ODE VII, the recommended order of choice of methods is as follows: Euler, Nystrom, Fourth Order Adams, Hamming, Third Order Adams, and lastly the Second Order Adams. The Milne and Hermite are not considered and should not be used for this particular type of ODE since they are both unstable.

## H. ODE VIII

Table 19 presents the true solution values for ODE VIII from x = 1.0 to x = 10.0. Data analysis from Tables 19-A and 19-B indicates that any of the eight P-C sets can be used to solve ODE VIII if no specific criterion for accuracy is needed, since each method yields good and valid numerical solutions. However, for purposes of comparison, the Hamming method clearly stands out to be the best in terms of accuracy with Milne coming in second. The Milne method provides the

TABLE 18-C

ODE at x = 10.0

ABSOLUTE ERROR FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII
Double Precision

| 1/h | $\Sigma_{HA}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | 0 | $-2.398169\cdot10^{-7}$ | $-5.968010\cdot10^{-10}$ | 0 |
| 64 | 0 | $-1.929706\cdot10^{-6}$ | $-1.07296\cdot10^{-9}$ | 0 |
| 32 | 0 | $-1.561650\cdot10^{-5}$ | $-1.87211\cdot10^{-9}$ | $-5.0\cdot10^{-14}$ |
| 16 | $5.665862\cdot10^{-8}$ | $-1.277970\cdot10^{-4}$ | $-3.04922\cdot10^{-9}$ | $-1.36\cdot10^{-12}$ |
| 8 | $5.374510\cdot10^{-8}$ | $-1.068471\cdot10^{-3}$ | $-1.119715\cdot10^{-8}$ | $1.715985\cdot10^{-7}$ |
| 4 | $2.632371\cdot10^{-7}$ | $-9.306166\cdot10^{-3}$ | $-3.729009\cdot10^{-7}$ | $-1.727044\cdot10^{-7}$ |
| 2 | $1.049942\cdot10^{-2}$ | $-8.771719\cdot10^{-2}$ | $-2.033756\cdot10^{-3}$ | $-9.570674\cdot10^{-3}$ |
| CPTS | 5.15 | 7.05 | 6.20 | 4.82 |

TABLE 19

TRUE SOLUTION VALUES FOR ODE VIII
Double Precision

| $x$ | $y_{exact}$ |
|---|---|
| 1.0 | 2.319776 |
| 2.0 | 2.482577 |
| 3.0 | 1.151562 |
| 4.0 | $4.691641 \cdot 10^{-1}$ |
| 5.0 | $3.833049 \cdot 10^{-1}$ |
| 6.0 | $7.562256 \cdot 10^{-1}$ |
| 7.0 | 1.928970 |
| 8.0 | 2.689507 |
| 9.0 | 1.510013 |
| 10.0 | $5.804096 \cdot 10^{-1}$ |

TABLE 19-A

ODE at x = 10.0

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-VI

Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | $1.391257 \cdot 10^{-6}$ | $-9.332 \cdot 10^{-11}$ | $7.611170 \cdot 10^{-7}$ | $1.595845 \cdot 10^{-7}$ |
| 64 | $6.532096 \cdot 10^{-6}$ | $-2.17509 \cdot 10^{-9}$ | $5.204737 \cdot 10^{-6}$ | $5.639547 \cdot 10^{-7}$ |
| 32 | $1.850381 \cdot 10^{-5}$ | $-5.498490 \cdot 10^{-8}$ | $1.836626 \cdot 10^{-5}$ | $4.499001 \cdot 10^{-6}$ |
| 16 | $7.305628 \cdot 10^{-5}$ | $-6.440709 \cdot 10^{-7}$ | $7.254094 \cdot 10^{-5}$ | $3.580536 \cdot 10^{-5}$ |
| 8 | $2.827282 \cdot 10^{-4}$ | $-4.145628 \cdot 10^{-6}$ | $2.733740 \cdot 10^{-4}$ | $2.841152 \cdot 10^{-4}$ |
| 4 | $9.532272 \cdot 10^{-4}$ | $-9.793708 \cdot 10^{-5}$ | $7.115245 \cdot 10^{-4}$ | $2.270258 \cdot 10^{-3}$ |
| 2 | $1.051152 \cdot 10^{-3}$ | $-2.707671 \cdot 10^{-3}$ | $-1.087262 \cdot 10^{-2}$ | $1.866206 \cdot 10^{-2}$ |
| CPTS | 6.14 | 4.63 | 5.11 | 6.48 |

TABLE 19-B

ODE at x = 10.0

ABSOLUTE ERROR FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII

Double Precision

| 1/h | $\Sigma_{HA}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | $-9.7 \cdot 10^{-13}$ | $9.902758 \cdot 10^{-7}$ | $2.367453 \cdot 10^{-7}$ | $-2.8983 \cdot 10^{-10}$ |
| 64 | $-4.057 \cdot 10^{-11}$ | $3.303524 \cdot 10^{-6}$ | $1.886624 \cdot 10^{-6}$ | $-5.53331 \cdot 10^{-9}$ |
| 32 | $-1.838 \cdot 10^{-9}$ | $7.843814 \cdot 10^{-6}$ | $1.498027 \cdot 10^{-5}$ | $-1.155997 \cdot 10^{-7}$ |
| 16 | $-1.006735 \cdot 10^{-7}$ | $-1.359301 \cdot 10^{-5}$ | $1.181971 \cdot 10^{-4}$ | $-1.563035 \cdot 10^{-6}$ |
| 8 | $-1.414250 \cdot 10^{-6}$ | $-4.482144 \cdot 10^{-4}$ | $9.248289 \cdot 10^{-4}$ | $-2.197783 \cdot 10^{-5}$ |
| 4 | $-7.603359 \cdot 10^{-5}$ | $-6.125271 \cdot 10^{-2}$ | $5.783513 \cdot 10^{-2}$ | $5.113170 \cdot 10^{-3}$ |
| 2 | $-9.956171 \cdot 10^{-4}$ | $-6.125271 \cdot 10^{-2}$ | $5.783513 \cdot 10^{-2}$ | $5.113170 \cdot 10^{-3}$ |
| CPTS | 5.31 | 7.61 | 8.11 | 5.09 |

least computing time with Hamming second. Thus for problems of the type ODE VIII, the Hamming method is the most highly recommended, followed by the Milne, Fourth Order Adams, Euler, Hermite, Nystrom, Third Order Adams, and lastly the Second Order Adams.

## I. ODE IX

True solution values are listed in Table 20 for ODE IX for the range of integration x = 1.0 to x = 10.0. From Tables 20-A and 20-B it was observed by analyzing the results that all eight methods provided valid numerical solutions. In terms of better accuracy on the average and the least computing time the Milne method is the best candidate with Hamming coming in next, followed by the Fourth Order Adams, Hermite, Third Order Adams, Euler, Nystrom and lastly the Second Order Adams. Thus if choice is to be made for solving classes of ODE belonging to ODE IX, the Milne method is highly recommended with Hamming as the next alternate.

## J. ODE X

This differential equation is an example of a controlled variable problem, where one variable is expressed as a function of x and substituted into the differential equation of the other variable to form a single first order ODE. The original two variable equations are:

$$y_1' = 1.38y_1 - 0.81y_2 \qquad (2\text{-}74)$$

$$y_2' = 2.16y_1 - 1.92y_2 \qquad (2\text{-}75)$$

TABLE 20

TRUE SOLUTION VALUES FOR ODE IX
Double Precision

| x | $y_{exact}$ |
|---|---|
| 1.0 | 2.069535 |
| 2.0 | 2.249705 |
| 3.0 | 4.179309 |
| 4.0 | 9.462527 |
| 5.0 | 10.633343 |
| 6.0 | 17.564095 |
| 7.0 | 42.421352 |
| 8.0 | 50.806493 |
| 9.0 | 74.608406 |
| 10.0 | 186.463649 |

TABLE 20-A

ODE IX at x = 10.0

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV

Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | $-1.591 \cdot 10^{-3}$ | $1.791 \cdot 10^{-8}$ | $-1.658 \cdot 10^{-3}$ | $1.403 \cdot 10^{-5}$ |
| 64 | $-6.435 \cdot 10^{-3}$ | $1.666 \cdot 10^{-7}$ | $-6.522 \cdot 10^{-3}$ | $1.173 \cdot 10^{-4}$ |
| 32 | $-2.578 \cdot 10^{-2}$ | $2.341 \cdot 10^{-6}$ | $-2.636 \cdot 10^{-2}$ | $9.226 \cdot 10^{-4}$ |
| 16 | $-1.025 \cdot 10^{-1}$ | $3.806 \cdot 10^{-5}$ | $-1.079 \cdot 10^{-1}$ | $7.170 \cdot 10^{-3}$ |
| 8 | $-4.022 \cdot 1-^{-1}$ | $1.016 \cdot 10^{-3}$ | $-4.569 \cdot 10^{-1}$ | $5.390 \cdot 10^{-2}$ |
| 4 | -1.450 | $4.234 \cdot 10^{-2}$ | -2.090 | $3.761 \cdot 10^{-1}$ |
| 2 | -3.737 | $9.791 \cdot 10^{-1}$ | -10.452 | 2.559 |
| CPTS | 6.99 | 5.40 | 7.09 | 6.42 |

TABLE 20-B

ODE IX at x = 10.0

ABSOLUTE ERROR FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII

Double Precision

| 1/h | $\Sigma_{HA}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | $-2.795 \cdot 10^{-9}$ | $-1.648 \cdot 10^{-3}$ | $6.166 \cdot 10^{-5}$ | $6.922 \cdot 10^{-8}$ |
| 64 | $-6.428 \cdot 10^{-8}$ | $-6.720 \cdot 10^{-3}$ | $4.827 \cdot 10^{-4}$ | $9.598 \cdot 10^{-7}$ |
| 32 | $-2.136 \cdot 10^{-6}$ | $-2.788 \cdot 10^{-2}$ | $3.693 \cdot 10^{-3}$ | $1.610 \cdot 10^{-5}$ |
| 16 | $-8.656 \cdot 10^{-5}$ | $-1.190 \cdot 10^{-1}$ | $2.690 \cdot 10^{-2}$ | $2.948 \cdot 10^{-4}$ |
| 8 | $-9.897 \cdot 10^{-4}$ | $-5.288 \cdot 10^{-1}$ | $1.755 \cdot 10^{-1}$ | $7.000 \cdot 10^{-3}$ |
| 4 | $-5.696 \cdot 10^{-2}$ | -2.433 | $8.959 \cdot 10^{-1}$ | $2.261 \cdot 10^{-1}$ |
| 2 | 2.108 | -10.866 | 3.597 | 4.380 |
| CPTS | 5.73 | 7.93 | 8.33 | 5.44 |

with initial conditions

$$y_1(0) = -2.9995 \qquad (2\text{-}76)$$

$$y_2(0) = 4.0010 \qquad (2\text{-}77)$$

and the analytic solutions given by

$$y_1 = 0.0005\, e^{-3x} - 3e^{-0.3x} \qquad (2\text{-}78)$$

$$y_2 = 0.001\, e^{-3x} + 4e^{-0.3x} \qquad (2\text{-}79)$$

Since the analytical solutions of both differential equations are known, it is possible to treat one variable as a function of x and substitute it to the differential equation of the other variable to form a single first order ODE which now can be solved by the P-C sets. Choosing $y_2$ as the known function given by (2-79) and substituting it in (2-74), the resulting equation is ODE X with initial condition given by (2-76) and exact solution by (2-78). Note that $y_2$ remains a function of the single independent variable x and as such changes as x changes. The other variable $y_1$ forms a first order differential equation in the standard form $y_1' = f(x,y)$. The point that is being illustrated here is that this concept can be applied to problems like rate of chemical reaction, falling bodies, aircraft or ballistic missiles flight wherein time variable is the most important consideration. All other variables can be expressed as functions of the single independent variable time and given constants or initial conditions. In such cases the problem can be reduced to a single differential equation and the methods considered herein can

be applied (time (t) variable as x). This concept does not discount the fact that these P-C methods can be extended to solve simultaneous differential equations with some modifications in the algorithms. However, since this extension is not relevant to the purpose of this particular paper in analyzing and comparing the predictor-corrector methods as applied to a variety of ODEs, it is not considered here, but will be mentioned as a further field of study.

Returning now to the solution of ODE X, Table 21 shows the true solution values from x = 1.0 to x = 10.0. From Table 21-A the Milne seemed to produce the best accuracy but by analyzing further the behavior of these errors in step-by-step integration, which is shown in Table 21-B, it is noted that the errors of the Milne method are increasing in a uniform fashion such that as the range of integration is increased the error grows while the true solution values as shown in Table 21 decrease. Thus eventually the error will overcome the solution. The same is true for the Hermite method, while for the Euler and the Nystrom the error decreases as the range of integration increases thus providing a valid numerical solution. The behavior of the Milne and Hermite methods is expected since for ODE X it is easy to see that $C < 0$, and as such the Milne and Hermite methods are unstable. Between the Euler and the Nystrom, the Euler method is an easy choice both in accuracy and computing time.

TABLE 21

TRUE SOLUTION VALUES FOR ODE X

Double Precision

| x | $y_{exact}$ |
|---|---|
| 1.0 | -2.222429 |
| 2.0 | -1.646433 |
| 3.0 | -1.219708 |
| 4.0 | $-9.035826\cdot10^{-1}$ |
| 5.0 | $-6.693904\cdot10^{-1}$ |
| 6.0 | $-4.958966\cdot10^{-1}$ |
| 7.0 | $-3.673692\cdot10^{-1}$ |
| 8.0 | $-2.721538\cdot10^{-1}$ |
| 9.0 | $-2.016165\cdot10^{-1}$ |
| 10.0 | $-1.493612\cdot10^{-1}$ |

TABLE 21-A

ODE at x = 10.0

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV

Double Precision

| 1/h | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 128 | $4.839775\cdot10^{-7}$ | $-1.205\cdot10^{-11}$ | $-3.684268\cdot10^{-8}$ | $-9.818242\cdot10^{-4}$ |
| 64 | $-9.887212\cdot10^{-8}$ | $-5.3459\cdot10^{-10}$ | $-8.584152\cdot10^{-8}$ | $-3.913399\cdot10^{-3}$ |
| 32 | $-4.893782\cdot10^{-7}$ | $-3.606138\cdot10^{-8}$ | $-3.832304\cdot10^{-7}$ | $-1.558160\cdot10^{-2}$ |
| 16 | $-1.149738\cdot10^{-6}$ | $-4.286345\cdot10^{-7}$ | $-1.793736\cdot10^{-6}$ | $-6.236379\cdot10^{-2}$ |
| 8 | $-3.757046\cdot10^{-6}$ | $-1.576153\cdot10^{-5}$ | $-4.903244\cdot10^{-6}$ | $-2.594684\cdot10^{-1}$ |
| 4 | $-9.800289\cdot10^{-8}$ | $-2.583136\cdot10^{-4}$ | $-2.093989\cdot10^{-5}$ | -1.288607 |
| 2 | $2.784363\cdot10^{-4}$ | $-8.436503\cdot10^{-3}$ | $-1.350331\cdot10^{-4}$ | -11.095378 |
| CPTS | 7.03 | 5.33 | 10.0 | 9.83 |

TABLE 21-B

ODE X with h = 0.5

ABSOLUTE ERROR FOR P-C-I, P-C-II, P-C-III, and P-C-IV
Double Precision

| x | $\Sigma_E$ | $\Sigma_M$ | $\Sigma_N$ | $\Sigma_{HE}$ |
|---|---|---|---|---|
| 1.0 | $2.187 \cdot 10^{-3}$ | | $-5.056 \cdot 10^{-4}$ | $-5.661 \cdot 10^{-2}$ |
| 2.0 | $2.565 \cdot 10^{-3}$ | $-3.821 \cdot 10^{-5}$ | $-1.101 \cdot 10^{-3}$ | $-8.576 \cdot 10^{-2}$ |
| 3.0 | $2.144 \cdot 10^{-3}$ | $-1.905 \cdot 10^{-4}$ | $-1.016 \cdot 10^{-3}$ | $-1.542 \cdot 10^{-1}$ |
| 4.0 | $1.651 \cdot 10^{-3}$ | $-3.765 \cdot 10^{-4}$ | $-7.984 \cdot 10^{-4}$ | $-2.833 \cdot 10^{-1}$ |
| 5.0 | $1.239 \cdot 10^{-3}$ | $-6.420 \cdot 10^{-4}$ | $-6.013 \cdot 10^{-4}$ | $-5.219 \cdot 10^{-1}$ |
| 6.0 | $9.223 \cdot 10^{-4}$ | $-1.076 \cdot 10^{-3}$ | $-4.475 \cdot 10^{-4}$ | $-9.619 \cdot 10^{-1}$ |
| 7.0 | $6.843 \cdot 10^{-4}$ | $-1.800 \cdot 10^{-3}$ | $-3.319 \cdot 10^{-3}$ | -1.772 |
| 8.0 | $5.072 \cdot 10^{-4}$ | $-3.012 \cdot 10^{-3}$ | $-2.460 \cdot 10^{-3}$ | -3.266 |
| 9.0 | $3.758 \cdot 10^{-4}$ | $-5.041 \cdot 10^{-3}$ | $-1.822 \cdot 10^{-3}$ | -6.020 |
| 10.0 | $2.784 \cdot 10^{-4}$ | $-8.436 \cdot 10^{-3}$ | $-1.350 \cdot 10^{-3}$ | -11.095 |

From Table 21-C, all the last four methods exhibited stable numerical solutions. For accuracy the Hamming method offers the best choice, with Fourth Order Adams coming next, though the computing time for Hamming is a little longer than the Fourth Order Adams. Therefore for problems in the class of ODE X, the order of recommended preference is: Hamming, Fourth Order Adams, Third Order Adams, Euler, Nystrom, and the Second Order Adams as the last choice. The Milne and Hermite methods should not be used for this particular type of problem.

## K. SUMMARY

For series of h values from $h = 2^{-7}$ to $h = 2^{-1}$ and range of integration up to $x = 10.0$, Tables 22, 22-A, and 22-B list the summary of results for the eight predictor-corrector sets considered, each run on the test ODEs I to X. From these comparative results, it is clearly established that the best numerical methods in order of decreasing efficiency in general are:

1. Hamming Modified P-C Set
2. Fourth Order Adams-Moulton P-C Set
3. Milne P-C Set
4. Third Order Adams-Moulton P-C Set
5. Euler P-C Set
6. Nystrom P-C Set
7. Second Order Adams-Moulton P-C Set
8. Hermite P-C Set.

TABLE 21-C

ODE X at x = 10.0

ABSOLUTE ERROR FOR P-C-V, P-C-VI, P-C-VII, and P-C-VIII
Double Precision

| 1/h | $\Sigma_{HA}$ | $\Sigma_{A2}$ | $\Sigma_{A3}$ | $\Sigma_{A4}$ |
|---|---|---|---|---|
| 128 | 0 | $-1.149665\cdot10^{-7}$ | $-9.07653\cdot10^{-9}$ | $-4.0\cdot10^{-14}$ |
| 64 | 0 | $-8.476336\cdot10^{-7}$ | $-7.319508\cdot10^{-8}$ | $-6.9\cdot10^{-13}$ |
| 32 | $9.0\cdot10^{-14}$ | $-6.538981\cdot10^{-6}$ | $-5.949594\cdot10^{-7}$ | $-1.380\cdot10^{-11}$ |
| 16 | $8.773099\cdot10^{-8}$ | $-5.207967\cdot10^{-5}$ | $-4.912304\cdot10^{-6}$ | $-3.0945\cdot10^{-10}$ |
| 8 | $-7.065215\cdot10^{-8}$ | $-4.272440\cdot10^{-4}$ | $-4.180375\cdot10^{-5}$ | $-2.697520\cdot10^{-8}$ |
| 4 | $4.17081\cdot10^{-9}$ | $-3.630456\cdot10^{-3}$ | $-3.757718\cdot10^{-4}$ | $1.553614\cdot10^{-7}$ |
| 2 | $-4.137219\cdot10^{-5}$ | $-3.165651\cdot10^{-2}$ | $-3.628554\cdot10^{-3}$ | $-1.293084\cdot10^{-6}$ |
| CPTS | 6.02 | 9.45 | 9.98 | 5.45 |

TABLE 22

SUMMARY OF RESULTS FOR ODE I TO ODE IV

FOR $h=2^{-7}$ TO $h=2^{-1}$ AT $x=10.0$

| ODE Number | Stable P-C Sets in Order of Preference | Average Accuracy | Computing Time in sec. | Unstable P-C Sets |
|---|---|---|---|---|
| I | 1. Hamming | $10^{-4}$ | 5.76 | 1. Milne |
| | 2. Fourth Order Adams | $10^{-3}$ | 5.99 | 2. Hermite |
| | 3. Third Order Adams | $10^{-3}$ | 8.55 | |
| | 4. Euler | $10^{-2}$ | 7.65 | |
| | 5. Nystrom | $10^{-2}$ | 7.46 | |
| | 6. Second Order Adams | $10^{-2}$ | 7.98 | |
| II | 1. Hamming | $10^{-4}$ | 5.26 | 1. Hermite |
| | 2. Fourth Order Adams | $10^{-3}$ | 4.96 | |
| | 3. Milne | $10^{-3}$ | 4.69 | |
| | 4. Euler | $10^{-1}$ | 6.38 | |
| | 5. Second Order Adams | $10^{-1}$ | 7.79 | |
| | 6. Third Order Adams | $10^{-1}$ | 8.36 | |
| | 7. Nystrom | $10^{-1}$ | 5.34 | |
| III | 1. Hamming | $10^{-8}$ | 6.23 | None |
| | 2. Milne | $10^{-7}$ | 5.59 | |
| | 3. Fourth Order Adams | $10^{-7}$ | 8.23 | |
| | 4. Third Order Adams | $10^{-5}$ | 10.0 | |
| | 5. Hermite | $10^{-4}$ | 7.92 | |
| | 6. Euler | $10^{-4}$ | 7.81 | |
| | 7. Second Order Adams | $10^{-4}$ | 10.0 | |
| | 8. Nystrom | $10^{-4}$ | 7.12 | |
| IV | 1. Hamming | $10^{-10}$ | 4.45 | None |
| | 2. Milne | $10^{-9}$ | 3.83 | |
| | 3. Fourth Order Adams | $10^{-8}$ | 4.02 | |
| | 4. Third Order Adams | $10^{-6}$ | 7.02 | |
| | 5. Nystrom | $10^{-5}$ | 3.67 | |
| | 6. Second Order Adams | $10^{-5}$ | 6.46 | |
| | 7. Euler | $10^{-5}$ | 4.39 | |
| | 8. Hermite | $10^{-4}$ | 6.74 | |

TABLE 22-A

SUMMARY OF RESULTS FOR ODE V TO ODE VIII

FOR $h=2^{-7}$ TO $h=2^{-1}$ AT $x=10.0$

| ODE Number | Stable P-C Sets in Order of Preference | Accuracy | Computing Time in sec. | Unstable P-C Sets |
|---|---|---|---|---|
| V | 1. Hamming | $10^{-13}$ | 6.31 | None |
| | 2. Milne | $10^{-12}$ | 5.58 | |
| | 3. Fourth Order Adams | $10^{-12}$ | 7.78 | |
| | 4. Third Order Adams | $10^{-11}$ | 10.0 | |
| | 5. Hermite | $10^{-10}$ | 5.35 | |
| | 6. Nystrom | $10^{-7}$ | 5.16 | |
| | 7. Euler | $10^{-7}$ | 5.63 | |
| | 8. Second Order Adams | $10^{-7}$ | 10.0 | |
| VI | 1. Hamming | $10^{-2}$ | 5.0 | 1. Euler |
| | 2. Fourth Order Adams | $10^{-1}$ | 4.79 | 2. Nystrom |
| | 3. Milne | $10^{-1}$ | 4.40 | 3. Hermite |
| | | | | 4. Third Order Adams |
| | | | | 5. Second Order Adams |
| VII | 1. Euler | $10^{-10}$ | 4.42 | 1. Milne |
| | 2. Nystrom | $10^{-9}$ | 4.14 | 2. Hermite |
| | 3. Fourth Order Adams | $10^{-8}$ | 4.82 | |
| | 4. Hamming | $10^{-8}$ | 5.15 | |
| | 5. Third Order Adams | $10^{-7}$ | 6.20 | |
| | 6. Second Order Adams | $10^{-4}$ | 7.05 | |
| VIII | 1. Hamming | $10^{-8}$ | 5.31 | None |
| | 2. Milne | $10^{-7}$ | 4.63 | |
| | 3. Fourth Order Adams | $10^{-6}$ | 5.09 | |
| | 4. Euler | $10^{-5}$ | 6.14 | |
| | 5. Hermite | $10^{-5}$ | 6.48 | |
| | 6. Nystrom | $10^{-4}$ | 5.11 | |
| | 7. Third Order Adams | $10^{-4}$ | 8.11 | |
| | 8. Second Order Adams | $10^{-4}$ | 7.61 | |

TABLE 22-B

SUMMARY OF RESULTS FOR ODE IX AND ODE X FOR $h=2^{-7}$ TO $h=2^{-1}$ AT $x=10.0$

| ODE Number | Stable P-C Sets in Order of Preference | Average Accuracy | Computing Time in secs. | Unstable P-C Sets |
|---|---|---|---|---|
| IX | 1. Milne | $10^{-4}$ | 5.40 | |
| | 2. Hamming | $10^{-4}$ | 5.73 | |
| | 3. Fourth Order Adams | $10^{-4}$ | 5.44 | |
| | 4. Hermite | $10^{-3}$ | 6.42 | |
| | 5. Third Order Adams | $10^{-3}$ | 8.33 | |
| | 6. Euler | $10^{-2}$ | 6.99 | |
| | 7. Nystrom | $10^{-2}$ | 7.09 | |
| | 8. Second Order Adams | $10^{-2}$ | 7.93 | |
| X | 1. Hamming | $10^{-10}$ | 6.02 | 1. Milne |
| | 2. Fourth Order Adams | $10^{-9}$ | 5.45 | 2. Hermite |
| | 3. Third Order Adams | $10^{-6}$ | 9.98 | |
| | 4. Euler | $10^{-6}$ | 7.03 | |
| | 5. Nystrom | $10^{-6}$ | 10.0 | |
| | 6. Second Order Adams | $10^{-5}$ | 9.45 | |

The most important criteria applied is accuracy followed by computing time. It is obvious that the Hamming method did not produce the least computing time as compared to the other methods especially with that of Milne and the Fourth Order Adams methods. This difference of less than a second in computing time is attributed to the fact that the Hamming method used additional computation for the truncation error to modify the predicted and corrected values for every iteration. Analyzing the computing time for each method within the range of their stability, the list below shows the order of increasing computing time:

| P-C Set | Average Computing Time |
|---|---|
| 1. Milne | 4.82 secs. |
| 2. Hamming | 5.60 secs. |
| 3. Nystrom | 5.65 secs. |
| 4. Fourth Order Adams | 6.11 secs. |
| 5. Euler | 6.19 secs. |
| 6. Hermite | 6.58 secs. |
| 7. Second Order Adams | 8.40 secs. |
| 8. Third Order Adams | 8.69 secs. |

This confirmed the idea that convergence of a corrector does not necessarily assure the convergence towards the true solution values, $y(x_n)$, but only to some definite value, $y_{n+1}$. This is best illustrated in the case of Hermite and Third Order Adams methods with 6.58 secs and 8.69 secs average computing time, respectively. By recalling from the order of efficiency of the P-C sets, the Third Order Adams ranks

fourth in accuracy while Hermite ranks eighth, which points to the fact that though the Hermite method converges much faster than the third order Adams, its accuracy is much poorer than the Third Order Adams. Clearly then it can be seen that the Hermite method is converging only to some definite value, $y_{n+1}$, but not to the true solution, $y(x_n)$, otherwise it should have much better accuracy than the Third Order Adams method, which took much longer to converge to a definite value $y_{n+1}$, in each iteration. This analysis is meaningful due to the fact that a standard mode of application is used, that of $P(EC)^2$. It should be noted however that rapid convergence is essential to accuracy as shown by the Hamming, Milne, and Fourth Order Adams methods which all rank well above the other methods in both accuracy and computing time in general.

# X. CONCLUSIONS

The following conclusions can be drawn and recommendations made from the analysis and numerical results obtained in this paper.

1. Numerical instability results from extraneous solutions of the difference equations which bear no connection to the exact solution. The conditions of asymptotic (strong) and absolute stability can be seen clearly by reference to the extraneous solutions. If in the limit $h \to 0$, the extraneous solutions vanish as $n \to \infty$, then the method is strongly stable and convergent. If, for values of h less than some $h_o$, the extraneous solutions vanish as $n \to \infty$, the method is absolutely stable. Relative stability is a significant concept for ODE where $f > 0$. The condition provides that extraneous solutions will not grow more rapidly or decay more slowly than the true solution. Thus, before a method is used, the characteristic roots of its related difference equation must be analyzed.

2. The finite real negative stability bounds for the P-C methods considered, as determined by numerical experiments, are in reasonable agreement with theoretical predictions. As such experimental bounds could be used as a guide to the proper selection of a method to solve a particular problem.

3. The stability of a P-C set depends on both the predictor and corrector equations. Thus, when two P-C sets have the

same corrector but different predictors, choose the one with a better predictor. Likewise, when two P-C sets use the same predictor with different correctors, choose the one with a better corrector.

4. P-C sets with higher order truncation errors are not necessarily better. In choosing a P-C set, the order of truncation error should not be the sole criterion.

5. The convergence of the corrector formula will ensure that the sequence of approximations will converge to some definite value but not necessarily to the true solution values. As such, the fast computation time of a method does not necessarily imply greater accuracy for the resulting numerical solution.

6. If the integration will involve a large number of steps, a stable method should be used.

7. If the function evaluation is lengthy, the range of integration large, and better accuracy and lesser cost of computing time are prime considerations, then the following P-C methods are recommended, based on the overall efficiency they have demonstrated in numerical experiments on a wide variety of different test ODEs:

   a) Hamming P-C Sets

   b) Fourth Order Adams P-C Set

   c) Milne P-C Set

   d) Third Order Adams P-C Set

   e) Euler P-C Set

   f) Nystrom P-C Set

g) Second Order Adams P-C Set

h) Hermite P-C Set

The listing indicates order of preference. However caution must be exercised not to use Milne and Hermite methods if $f_y < 0$.

8. For maximum accuracy to be achieved, select a step size h, for which the truncation error and the roundoff error have equal orders of magnitude.

9. If high accuracy results with less roundoff error are desired, the procedure should use double precision, which involves only slightly more computing time than single precision in an IBM 360/67 computer.

# XI. EXTENSIONS

The following topics, which are direct extensions of this paper, are interesting areas of further research and study in the wide field of numerical analysis:

1. Stability versus Accuracy

A necessary condition for numerical integration is stability; that is, a stable value of h must be employed when a finite stability boundary exists. Nevertheless, there is no guarantee that all values of h from zero to the limiting value will yield accurate results. Thus the question that must be faced is: Of what value is a method that is stable in a region where it is inaccurate? The answer to this question will show that the relationship between stability and accuracy is fundamental to the choice of h for a particular method.

2. Increasing the Stability Bounds

It has been shown that in some methods the real stability bound is somewhat constrained. Two possible ways where the stability bounds can be increased are:

a) By the use of a weighting factor which involves experimenting with different values of the free parameters that are used in the method by undetermined coefficients, similar to Hamming's corrector formula derivation.

b) By averaging, which means computing the new value of $y_{n+1}$ after a finite number of steps (say fifty) of a particular

method, as the average of the old $y_{n+1}$ and a value called $y_{n+1}$ generated by another method. This should be done periodically.

3. Predictor-Corrector Methods Interaction

The interaction of the different methods in solving a particular problem involves use of the P-C sets as subroutines, wherein one method will be used to provide the solution for smaller values of h, then another method will be used to solve for larger values of h. This should be quite interesting since it has been shown that some methods exhibit better accuracy for small values of h, then become inaccurate for large values of h, while other methods maintain their accuracy up to large values of h but might be prohibitive to use for small values of h due to their longer computing time.

4. Solution of Systems of Differential Equations

Revision of the algorithms to enable the solution of systems of differential equations and higher order differential equations. This extension is straightforward. It involves only extra computational steps using the same formulas. Complexity arises from the need to use vectors and arrays for temporary storage.

5. Practical Applications

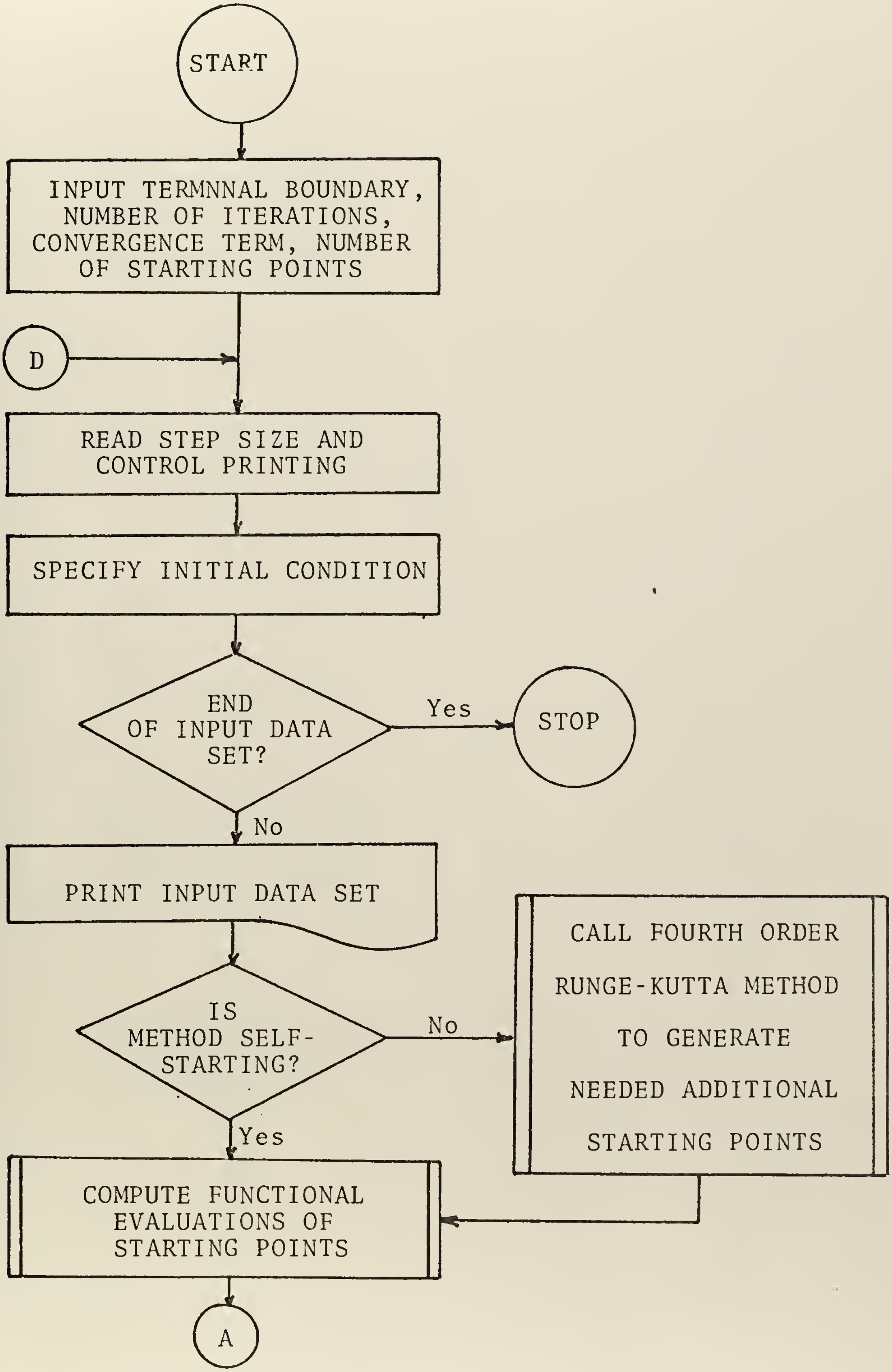
Applications to real-life problems such as heat flow problems, simple electrical circuits, force problems, rate of bacterial growth, rate of decomposition of radioactive

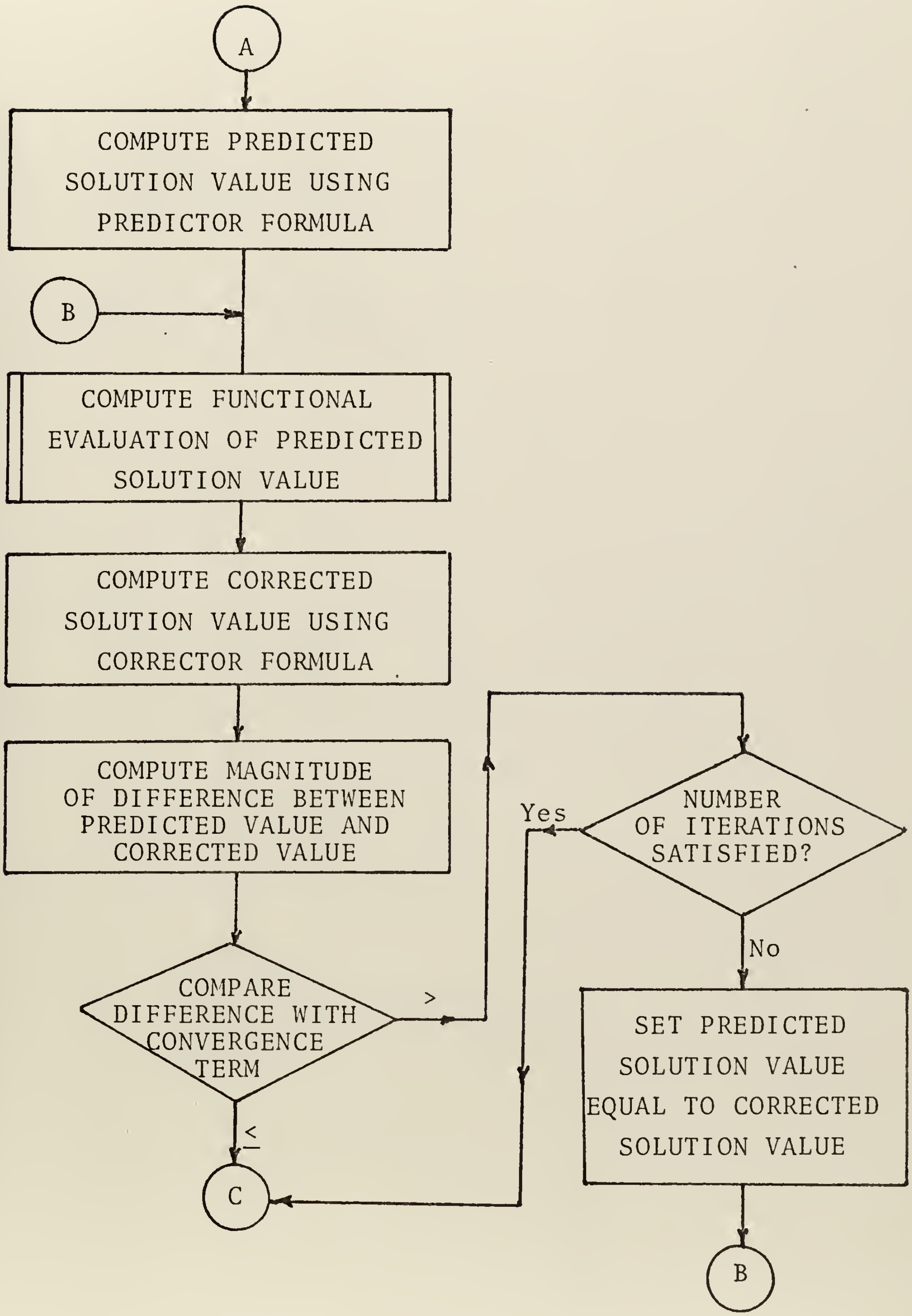
material, crystallization rate of a chemical compound, rate of population growth, and so forth.

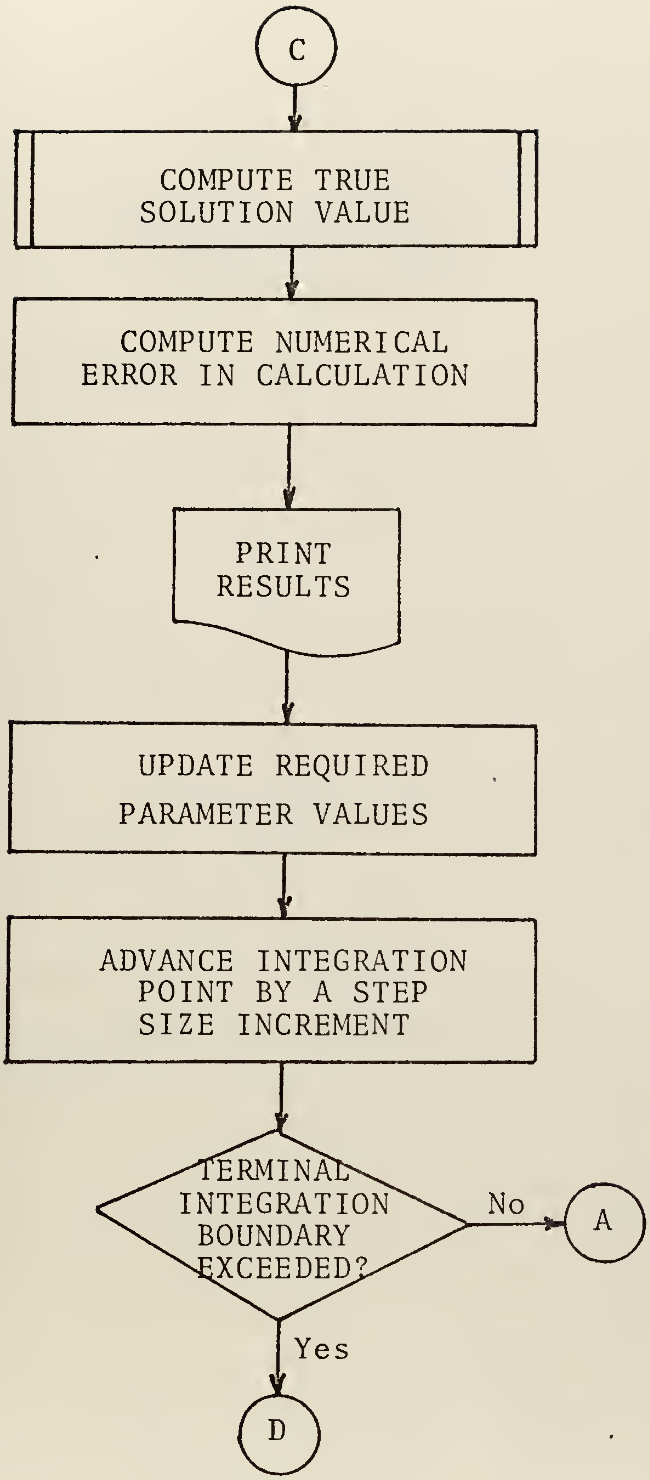
6. Adjustment of the Step Size During the P-C Solution

This involves monitoring the modified truncation error developed by Hamming. If the value is too large, then the step size is too large, and the calculation should be repeated with a smaller value of h, say h/2. Note that it is not necessary to go back to the beginning of the calculation but only to the point at which the truncation error becomes too large. If the truncation error is too small, computation time is being wasted and h should be increased, say to 2h.

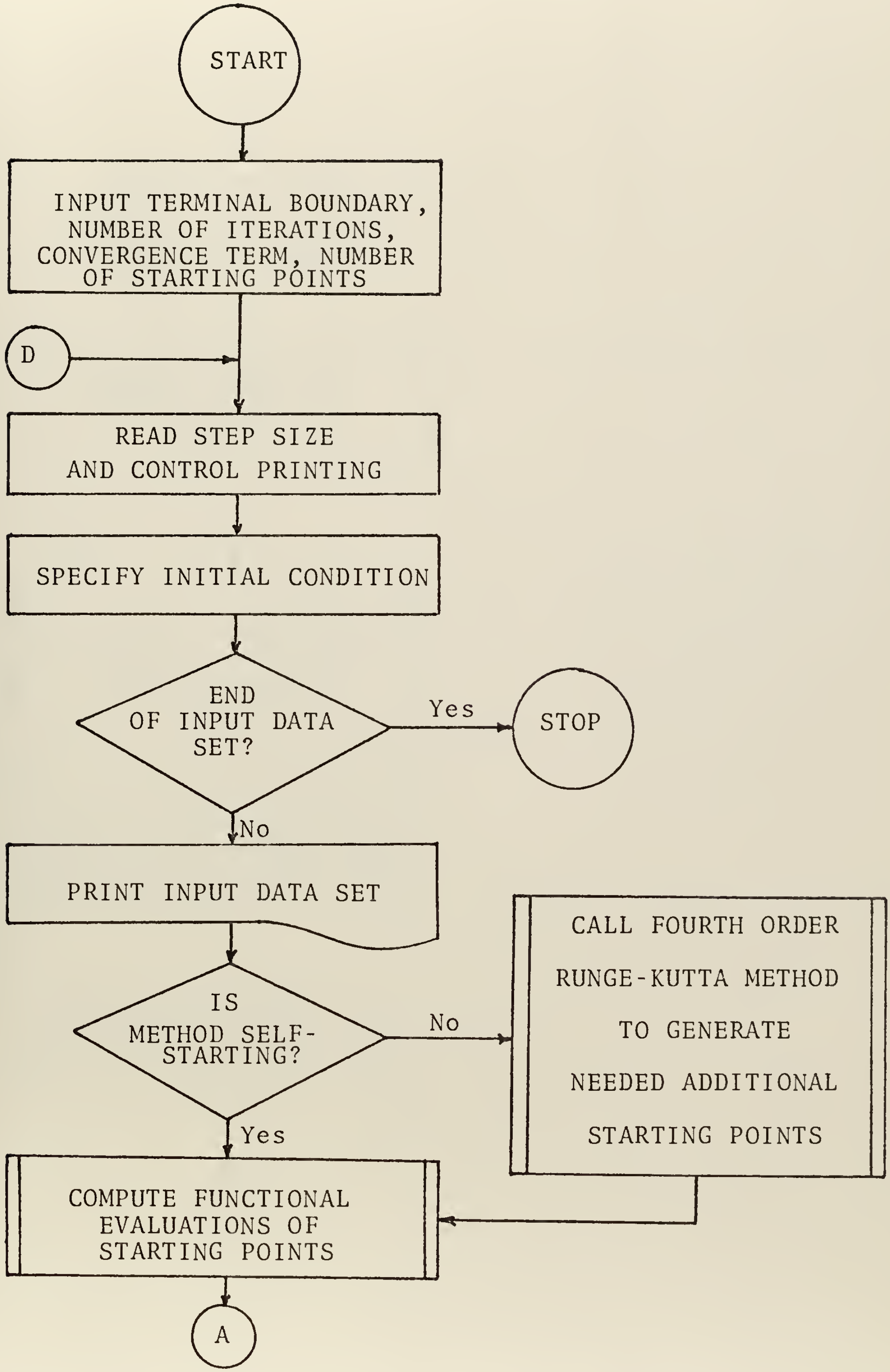
FLOW CHART FOR ALGORITHM 1

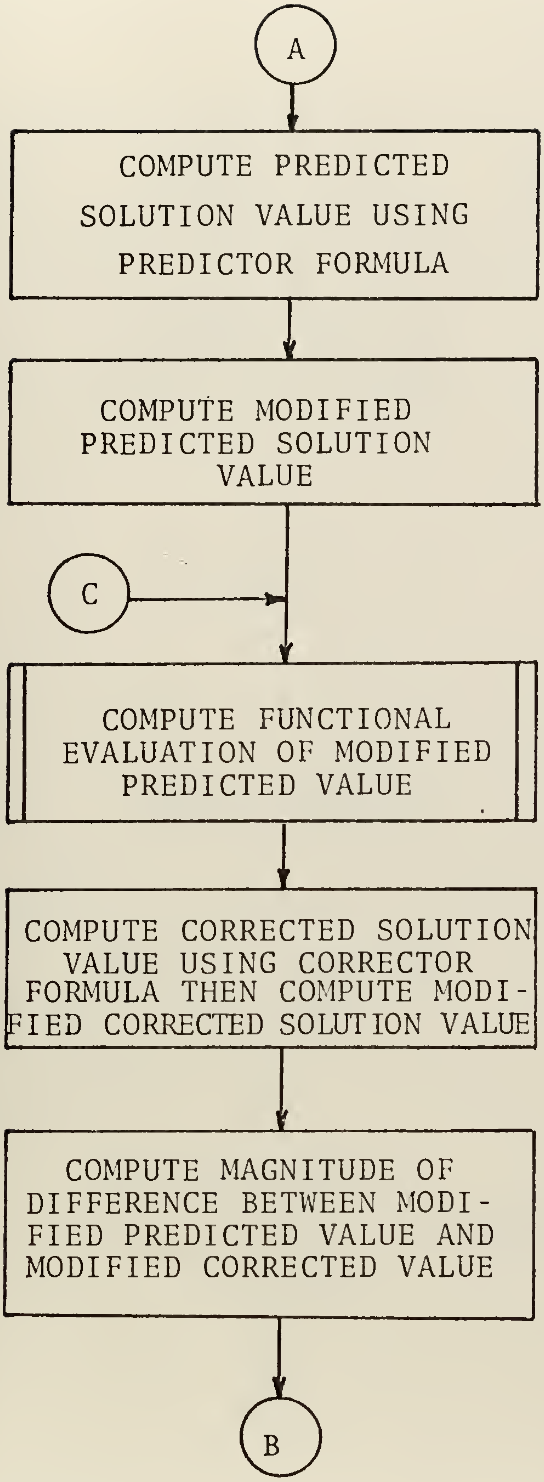


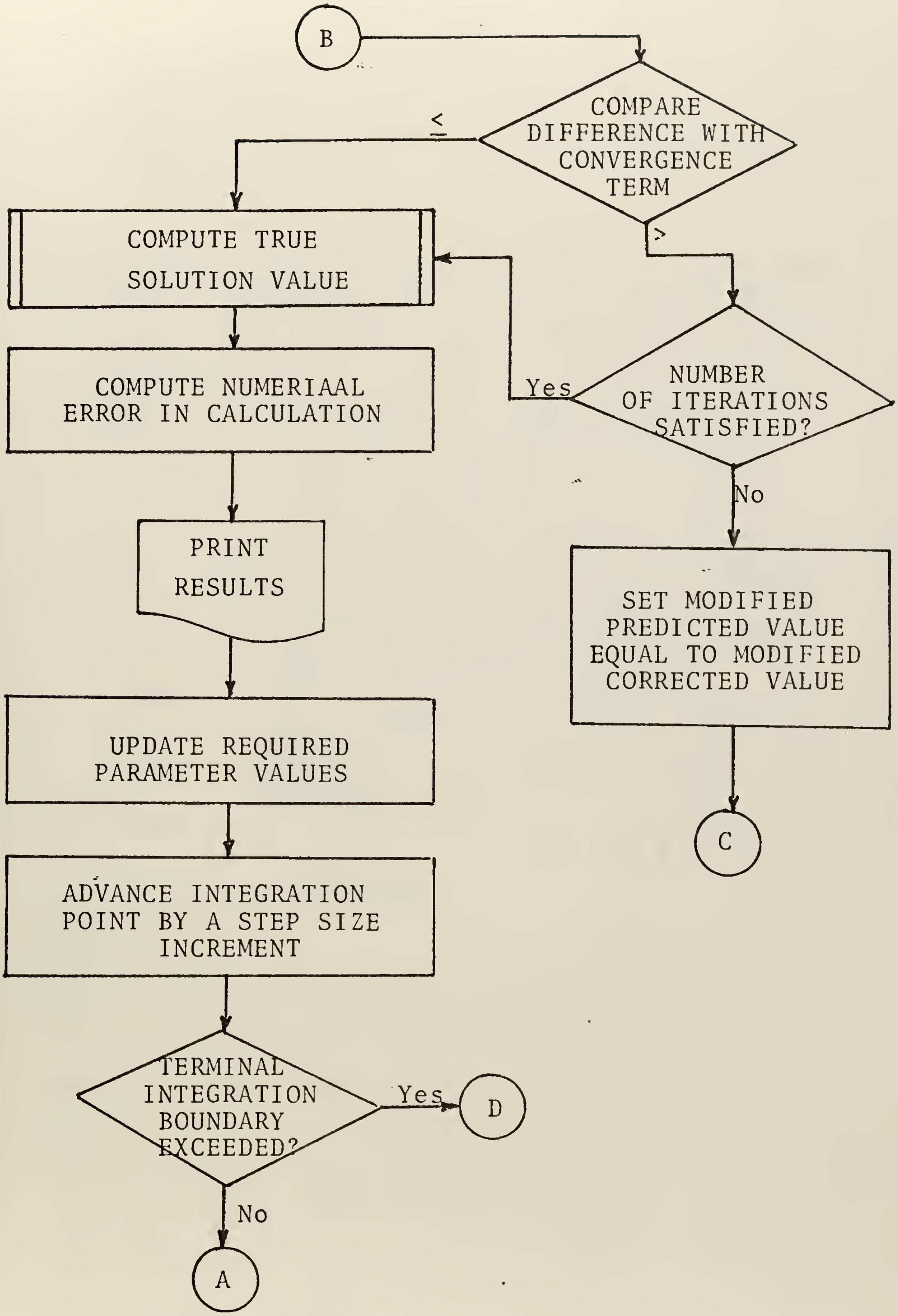

A
COMPUTE PREDICTED SOLUTION VALUE USING PREDICTOR FORMULA
B
COMPUTE FUNCTIONAL EVALUATION OF PREDICTED SOLUTION VALUE
COMPUTE CORRECTED SOLUTION VALUE USING CORRECTOR FORMULA
COMPUTE MAGNITUDE OF DIFFERENCE BETWEEN PREDICTED VALUE AND CORRECTED VALUE
COMPARE DIFFERENCE WITH CONVERGENCE TERM
>
≤
C
NUMBER OF ITERATIONS SATISFIED?
Yes
No
SET PREDICTED SOLUTION VALUE EQUAL TO CORRECTED SOLUTION VALUE
B


C
COMPUTE TRUE
SOLUTION VALUE
COMPUTE NUMERICAL
ERROR IN CALCULATION
PRINT
RESULTS
UPDATE REQUIRED
PARAMETER VALUES
ADVANCE INTEGRATION
POINT BY A STEP
SIZE INCREMENT
TERMINAL
INTEGRATION
BOUNDARY
EXCEEDED?
No
A
Yes
D

FLOW CHART FOR ALGORITHM 2




A
COMPUTE PREDICTED SOLUTION VALUE USING PREDICTOR FORMULA
COMPUTE MODIFIED PREDICTED SOLUTION VALUE
C
COMPUTE FUNCTIONAL EVALUATION OF MODIFIED PREDICTED VALUE
COMPUTE CORRECTED SOLUTION VALUE USING CORRECTOR FORMULA THEN COMPUTE MODIFIED CORRECTED SOLUTION VALUE
COMPUTE MAGNITUDE OF DIFFERENCE BETWEEN MODIFIED PREDICTED VALUE AND MODIFIED CORRECTED VALUE
B


B
COMPARE DIFFERENCE WITH CONVERGENCE TERM
≤
>
COMPUTE TRUE SOLUTION VALUE
NUMBER OF ITERATIONS SATISFIED?
Yes
No
COMPUTE NUMERIAAL ERROR IN CALCULATION
PRINT RESULTS
SET MODIFIED PREDICTED VALUE EQUAL TO MODIFIED CORRECTED VALUE
C
UPDATE REQUIRED PARAMETER VALUES
ADVANCE INTEGRATION POINT BY A STEP SIZE INCREMENT
TERMINAL INTEGRATION BOUNDARY EXCEEDED?
Yes
D
No
A

PROGRAM ONE

EULER PREDICTOR-CORRECTOR METHOD. ODE: Y'=-Y WITH Y(0)=1. TO SOLVE ANOTHER ODE SIMPLY CHANGE INITIAL CONDITIONS AND FUNCTION SUBROUTINES.

```
C       ******************************************************
C       * FORTRAN PROGRAM FOR EULER PREDICTOR-CORRECTOR      *
C       * METHOD IN THE NUMERICAL SOLUTION OF THE ORDINARY   *
C       * DIFFERENTIAL EQUATION IN THE FORM OF DY/DX=F(X,Y)  *
C       * WITH THE INITIAL CONDITION Y(X0)=Y0                *
C       * THE PARAMETERS TO THE PROGRAM HAVE THE FOLLOWING   *
C       * MEANINGS...                                        *
C       *    XMAX______THE TERMINAL BOUNDARY CONDITION       *
C       *    EPS_______THE CONVERGENCE TEST CONSTANT         *
C       *    MAX_______THE MAXIMUM NUMBER OF ITERATIONS      *
C       *    YP________THE PREDICTED VALUE OF THE NEXT       *
C       *              SOLUTION POINT                        *
C       *    YC________THE CORRECTED VALUE OF THE NEXT       *
C       *              SOLUTION POINT                        *
C       *    FP________THE PREDICTED FUNCTION EVALUATION OF YP*
C       *    FC________THE CORRECTED FUNCTION EVALUATION OF YC*
C       *    H_________THE STEP SIZE TO BE USED TO ADVANCE   *
C       *              THE POINT OF INTEGRATION              *
C       *    ICON______THE DESIRED POINT OF PRINTING THE     *
C       *              COMPUTED SOLUTION FOR EVERY FIXED     *
C       *              NUMBER OF INTEGRATION POINTS          *
C       *    YEXACT____THE TRUE SOLUTION VALUES              *
C       *    ERROR_____THE ABSOLUTE DEVIATION OF THE NUMERICAL*
C       *              SOLUTION FROM THE TRUE SOLUTION       *
C       *    FCT_______FUNCTION SUBROUTINE TO EVALUATE ODE   *
C       *              FUNCTION                              *
C       *    EXACT_____FUNCTION SUBROUTINE TO COMPUTE TRUE   *
C       *              SOLUTION VALUES                       *
C       ******************************************************
C
C
C       INITIALIZE INPUT CONSTANTS
C
        MAX=2
        XMAX=10.0
        EPS=0.000001
C
C       READ STEP SIZE AND CONTROL PRINTING
C
 5      READ(5,100) H,ICON
C
C       SPECIFY INITIAL CONDITION
C
        X0=0.0
        Y0=1.0
C
C       TEST FOR END OF INPUT DATA
C
        IF(ICON-0) 45,45,7
C
C       PRINT INPUT DATA SET
C
 7      WRITE(6,1000) H,ICON
        WRITE(6,1001) X0,Y0
C
C       COMPUTE NEXT SOLUTION POINT
C
        NC=(XMAX+H/2.)/H
        FP=FCT(X0,Y0)
        DO 40 N=1,NC
```

```
      M=1
      X0=X0+H
      YP=Y0+H*FP
 11   FC=FCT(X0,YP)
      YC=Y0+H/2.*(FP+FC)
      DELY=ABS(YC-YP)
C
C     TEST FOR CONVERGENCE
C
      IF(DELY-EPS) 30,30,15
C
C     TEST IF MAXIMUM NUMBER OF ITERATIONS IS SATISFIED
C
 15   IF(M-MAX) 20,20,30
 20   YP=YC
      M=M+1
      GO TO 11
C
C     COMPUTE TRUE SOLUTION VALUE
C
 30   YEXACT=EXACT(X0)
C
C     COMPUTE ERROR IN CALCULATION
C
      ERROR=YEXACT-YC
C
C     TEST IF DESIRED POINT OF PRINTING IS REACHED
C
      IF((N/ICON*ICON).EQ.N) WRITE(6,1002)X0,YP,YC,YEXACT,
     1ERROR
C
C     UPDATE REQUIRED PARAMETER VALUES
C
      Y0=YC
      FP=FC
 40   CONTINUE
C
C     LOOP BACK TO READ NEXT SET OF INPUT DATA
C
      GO TO 5
 45   STOP
 100  FORMAT(F10.7,I10)
 1000 FORMAT(///41X,'*INPUT DATA SET USED*'//43X,'H    =',
     1F10.7/43X,'ICON =',I10)
 1001 FORMAT(//29X,'**RESULTS OF EULER PREDICTOR-CORRECTOR
     1METHOD**'//10X,'    X    ',5X,'    YP    ',5X,'    YC    ',
     15X,'   YEXACT    ',5X,'    ERROR    '//12X,F10.7,44X,
     1F10.7)
 1002 FORMAT('0',9X,F15.7,2X,F15.7,2X,F15.7,2X,F15.7,2X,
     1F15.7)
      END


      FUNCTION FCT(XN,YN)
      FCT=-YN
      RETURN
      END


      FUNCTION EXACT(XX)
      EXACT=EXP(-XX)
      RETURN
      END
```

*INPUT DATA SET USED*

H = 0.5000000

ICON = 2

**RESULTS OF EULER PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 1.0000000 | 0.3588867 | 0.3634033 | 0.3678794 | 0.0044761 |
| 2.0000000 | 0.1309212 | 0.1325720 | 0.1353353 | 0.0027633 |
| 3.0000000 | 0.0477613 | 0.0483635 | 0.0497871 | 0.0014236 |
| 4.0000000 | 0.0174238 | 0.0176435 | 0.0183156 | 0.0006722 |
| 5.0000000 | 0.0063564 | 0.0064365 | 0.0067379 | 0.0003014 |
| 6.0000000 | 0.0023189 | 0.0023481 | 0.0024788 | 0.0001307 |
| 7.0000000 | 0.0008459 | 0.0008566 | 0.0009119 | 0.0000553 |
| 8.0000000 | 0.0003086 | 0.0003125 | 0.0003355 | 0.0000230 |
| 9.0000000 | 0.0001126 | 0.0001140 | 0.0001234 | 0.0000094 |
| 10.0000000 | 0.0000411 | 0.0000416 | 0.0000454 | 0.0000038 |

*INPUT DATA SET USED*

H = 1.0000000

ICON = 1

**RESULTS OF EULER PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 1.0000000 | 0.2500000 | 0.3750000 | 0.3678794 | -0.0071206 |
| 2.0000000 | 0.1562500 | 0.1718750 | 0.1353353 | -0.0365397 |
| 3.0000000 | 0.0507813 | 0.0683594 | 0.0497871 | -0.0185723 |
| 4.0000000 | 0.0258789 | 0.0300293 | 0.0183156 | -0.0117137 |
| 5.0000000 | 0.0095825 | 0.0122986 | 0.0067379 | -0.0055606 |
| 6.0000000 | 0.0044327 | 0.0052910 | 0.0024788 | -0.0028122 |
| 7.0000000 | 0.0017519 | 0.0021987 | 0.0009119 | -0.0012868 |
| 8.0000000 | 0.0007731 | 0.0009362 | 0.0003355 | -0.0006007 |
| 9.0000000 | 0.0003156 | 0.0003919 | 0.0001234 | -0.0002685 |
| 10.0000000 | 0.0001361 | 0.0001660 | 0.0000454 | -0.0001206 |

PROGRAM TWO

MILNE PREDICTOR-CORRECTOR METHOD. ODE: Y'=-Y WITH Y(0)=1. TO SOLVE ANOTHER ODE SIMPLY CHANGE INITIAL CONDITIONS AND FUNCTION SUBROUTINES.

```
C       **************************************************************
C       * FORTRAN PROGRAM FOR MILNE PREDICTOR-CORRECTOR              *
C       * METHOD IN THE NUMERICAL SOLUTION OF THE ORDINARY           *
C       * DIFFERENTIAL EQUATION IN THE FORM OF DY/DX=F(X,Y)          *
C       * WITH THE INITIAL CONDITION Y(XO)=YO                        *
C       * THE PARAMETERS TO THE PROGRAM HAVE THE FOLLOWING           *
C       * MEANINGS...                                                *
C       *    XMAX_______THE TERMINAL BOUNDARY CONDITION              *
C       *    EPS________THE CONVERGENCE TEST CONSTANT                *
C       *    MAX________THE MAXIMUM NUMBER OF ITERATIONS             *
C       *    YP_________THE PREDICTED VALUE OF THE NEXT              *
C       *               SOLUTION POINT                               *
C       *    YC_________THE CORRECTED VALUE OF THE NEXT              *
C       *               SOLUTION POINT                               *
C       *    FP_________THE FUNCTION EVALUATION                      *
C       *    H__________THE STEP SIZE TO BE USED TO ADVANCE          *
C       *               THE POINT OF INTEGRATION                     *
C       *    ICON_______THE DESIRED POINT OF PRINTING THE            *
C       *               COMPUTED SOLUTION FOR EVERY FIXED            *
C       *               NUMBER OF INTEGRATION POINTS                 *
C       *    YEXACT_____THE TRUE SOLUTION VALUES                     *
C       *    ERROR______THE ABSOLUTE DEVIATION OF THE NUMERICAL*
C       *               SOLUTION FROM THE TRUE SOLUTION              *
C       *    FCT________FUNCTION SUBROUTINE TO EVALUATE ODE          *
C       *               FUNCTION                                     *
C       *    EXACT______FUNCTION SUBROUTINE TO COMPUTE TRUE          *
C       *               SOLUTION VALUES                              *
C       *    RKUTTA_____THE SUBROUTINE USED TO GENERATE              *
C       *               NEEDED STARTING VALUES                       *
C       **************************************************************
C
C
C       INITIALIZE INPUT CONSTANTS
C
        XMAX=10.0
        MAX=2
        EPS=0.000001
        K=3
C
C       READ STEP SIZE AND CONTROL PRINTING
C
  5     READ(5,100) H,ICON
C
C       SPECIFY INITIAL CONDITION
C
        X0=0.0
        Y0=1.0
        YR=Y0
C
C       TEST FOR END OF INPUT DATA
C
        IF(ICON-0) 45,45,7
C
C       PRINT INPUT DATA SET
C
  7     WRITE(6,1000) H,ICON
        WRITE(6,1001) X0,Y0
C
C       COMPUTE NEXT SOLUTION POINT
```

```
      NC=(XMAX+H/2.)/H
      X1=X0+H
      X2=X0+2.*H
      X3=X0+3.*H
      X=X0+4.*H
      CALL RKUTTA(X0,YR,Y1,Y2,Y3,K,H)
      F1=FCT(X1,Y1)
      F2=FCT(X2,Y2)
      F3=FCT(X3,Y3)
      DO 40 N=4,NC
      M=1
      YP=Y0+4.*H/3.*(2.*F1-F2+2.*F3)
      FP=FCT(X,YP)
 11   YC=Y2+H/3.*(F2+4.*F3+FP)
      DELY=ABS(YC-YP)
C
C     TEST FOR CONVERGENCE
C
      IF(DELY-EPS) 30,30,15
C
C     TEST IF MAXIMUM NUMBER OF ITERATIONS IS SATISFIED
C
 15   IF(M-MAX) 20,20,30
 20   FP=FCT(X,YC)
      M=M+1
      GO TO 11
C
C     COMPUTE TRUE SOLUTION VALUE
C
 30   YEXACT=EXACT(X)
C
C     COMPUTE ERROR IN CALCULATION
C
      ERROR=YEXACT-YC
C
C     TEST IF DESIRED POINT OF PRINTING IS REACHED
C
      IF((N/ICON*ICON).EQ.N) WRITE(6,1002) X,YP,YC,YEXACT,
     1ERROR
C
C     UPDATE REQUIRED PARAMETER VALUES
C
      F1=F2
      F2=F3
      F3=FP
      Y0=Y1
      Y1=Y2
      Y2=Y3
      Y3=YC
      X=X+H
 40   CONTINUE
C
C     LOOP BACK TO READ NEXT SET OF INPUT DATA
C
      GO TO 5
 45   STOP
 100  FORMAT(F10.7,I10)
 1000 FORMAT(////41X,'*INPUT DATA SET USED*'//43X,'H    =',
     1F10.7/43X,'ICON =',I10)
 1001 FORMAT(//29X,'**RESULTS OF MILNE PREDICTOR-CORRECTOR
     1METHOD**'//10X,'    X    ',5X,'    YP    ',5X,'    YC    ',
     15X,'   YEXACT    ',5X,'    ERROR    '//12X,F10.7,44X,
     1F10.7)
 1002 FORMAT('0',9X,F15.7,2X,F15.7,2X,F15.7,2X,F15.7,2X,
     1F15.7)
      END
```

```
      SUBROUTINE RKUTTA(XX,YY,Y1,Y2,Y3,KK,HH)
      NN=1
17    CK1=HH*FCT(XX,YY)
      CK2=HH*FCT(XX+HH/2.0,YY+CK1/2.0)
      CK3=HH*FCT(XX+HH/2.0,YY+CK2/2.0)
      CK4=HH*FCT(XX+HH,YY+CK3)
      GO TO (101,102,103),NN
101   Y1=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y1
77    XX=XX+HH
      YEXACT=EXACT(XX)
      ERROR=YEXACT-YY
      WRITE(6,1004) XX,YY,YEXACT,ERROR
      NN=NN+1
      IF(NN-KK) 17,17,99
102   Y2=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y2
      GO TO 77
103   Y3=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y3
      GO TO 77
1004  FORMAT('0',9X,F15.7,2X,F15.7,16X,F15.7,2X,F15.7)
99    RETURN
      END

      FUNCTION FCT(XN,YN)
      FCT=-YN
      RETURN
      END

      FUNCTION EXACT(XK)
      EXACT=EXP(-XK)
      RETURN
      END
```

*INPUT DATA SET USED*

H = 0.5000000

ICON = 2

**RESULTS OF MILNE PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 0.5000000 | 0.6067709 | | 0.6065306 | -0.0002403 |
| 1.0000000 | 0.3681710 | | 0.367894 | -0.0002916 |
| 1.5000000 | 0.2233955 | | 0.2231301 | -0.0002654 |
| 2.0000000 | 0.1385594 | 0.1353100 | 0.1353353 | 0.0000253 |
| 3.0000000 | 0.0509259 | 0.0496315 | 0.0497871 | 0.0001555 |
| 4.0000000 | 0.0183159 | 0.0181091 | 0.0183156 | 0.0002066 |
| 5.0000000 | 0.0061856 | 0.0064749 | 0.0067379 | 0.0002630 |
| 6.0000000 | 0.0015049 | 0.0021262 | 0.0024788 | 0.0003525 |
| 7.0000000 | -0.0005340 | 0.0004233 | 0.0009119 | 0.0004886 |
| 8.0000000 | -0.0017356 | -0.0003513 | 0.0003355 | 0.0006868 |
| 9.0000000 | -0.0028179 | -0.0008470 | 0.0001234 | 0.0009704 |
| 10.0000000 | -0.0041233 | -0.0013280 | 0.0000454 | 0.0013734 |

*INPUT DATA SET USED*

H = 1.0000000

ICON = 1

**RESULTS OF MILNE PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 1.0000000 | 0.3750000 | | 0.3678794 | -0.0071206 |
| 2.0000000 | 0.1406250 | | 0.1353353 | -0.0052897 |
| 3.0000000 | 0.0527344 | | 0.0497871 | -0.0029473 |
| 4.0000000 | 0.0468752 | 0.0164931 | 0.0183156 | 0.0018226 |
| 5.0000000 | 0.0147570 | 0.0051922 | 0.0067379 | 0.0015457 |
| 6.0000000 | 0.0102883 | 0.0002441 | 0.0024788 | 0.0022346 |
| 7.0000000 | 0.0014462 | 0.0005433 | 0.0009119 | 0.0003686 |
| 8.0000000 | -0.0005506 | -0.0009222 | 0.0003355 | 0.0012576 |
| 9.0000000 | -0.0039288 | 0.0010040 | 0.0001234 | -0.0008806 |
| 10.0000000 | -0.0064989 | -0.0017257 | 0.0000454 | 0.0017711 |

PROGRAM THREE

NYSTROM PREDICTOR-CORRECTOR METHOD. ODE: Y'=-Y WIYH Y(0)=1. TO SOLVE ANOTHER ODE SIMPLY CHANGE INITIAL CONDITIONS AND FUNCTION SUBROUTINES.

```
C     ***************************************************************
C     * FORTRAN PROGRAM FOR NYSTROM PREDICTOR-CORRECTOR             *
C     * METHOD IN THE NUMERICAL SOLUTION OF THE ORDINARY            *
C     * DIFFERENTIAL EQUATION IN THE FORM OF DY/DX=F(X,Y)           *
C     * WITH THE INITIAL CONDITION Y(XO)=YO                         *
C     * THE PARAMETERS TO THE PROGRAM HAVE THE FOLLOWING            *
C     * MEANINGS...                                                 *
C     *   XMAX_______THE TERMINAL BOUNDARY CONDITION                *
C     *   EPS________THE CONVERGENCE TEST CONSTANT                  *
C     *   MAX________THE MAXIMUM NUMBER OF ITERATIONS               *
C     *   YP_________THE PREDICTED VALUE OF THE NEXT                *
C     *              SOLUTION POINT                                 *
C     *   YC_________THE CORRECTED VALUE OF THE NEXT                *
C     *              SOLUTION POINT                                 *
C     *   FP_________THE FUNCTION EVALUATION                        *
C     *   H__________THE STEP SIZE TO BE USED TO ADVANCE            *
C     *              THE POINT OF INTEGRATION                       *
C     *   ICON_______THE DESIRED POINT OF PRINTING THE              *
C     *              COMPUTED SOLUTION FOR EVERY FIXED              *
C     *              NUMBER OF INTEGRATION POINTS                   *
C     *   YEXACT_____THE TRUE SOLUTION VALUES                       *
C     *   ERROR______THE ABSOLUTE DEVIATION OF THE NUMERICAL*
C     *              SOLUTION FROM THE TRUE SOLUTION                *
C     *   FCT________FUNCTION SUBROUTINE TO EVALUATE ODE            *
C     *              FUNCTION.                                      *
C     *   EXACT______FUNCTION SUBROUTINE TO COMPUTE TRUE            *
C     *              SOLUTION VALUES                                *
C     *   RKUTTA_____THE SUBROUTINE USED TO GENERATE                *
C     *              NEEDED STARTING VALUES                         *
C     ***************************************************************
C
C
C
C     INITIALIZE INPUT CONSTANTS
C
      XMAX=10.0
      MAX=2
      EPS=0.000001
C
C     READ STEP SIZE AND CONTROL PRINTING
C
    5 READ(5,100) H,ICON
C
C     SPECIFY INITIAL CONDITION
C
      X0=0.0
      Y0=1.0
      YR=Y0
C
C     TEST FOR END OF INPUT DATA
C
      IF(ICON-0) 45,45,7
C
C     PRINT INPUT DATA SET
C
    7 WRITE(6,1000) H,ICON
      WRITE(6,1001) X0,Y0
C
C     COMPUTE NEXT SOLUTION POINT
C
```

```
      NC=(XMAX+H/2.)/H
      X1=X0+H
      X=X0+2.*H
      CALL RKUTTA(X0,YR,Y1,H)
      F1=FCT(X1,Y1)
      DO 40 N=2,NC
      M=1
      YP=Y0+2.*H*F1
      FP=FCT(X,YP)
 11   YC=Y1+H/2.*(F1+FP)
      DELY=ABS(YC-YP)
C
C     TEST FOR CONVERGENCE
C
      IF(DELY-EPS) 30,30,15
C
C     TEST IF MAXIMUM NUMBER OF ITERATIONS IS SATISFIED
C
 15   IF(M-MAX) 20,20,30
 20   FP=FCT(X,YC)
      M=M+1
      GO TO 11
C
C     COMPUTE TRUE SOLUTION VALUE
C
 30   YEXACT=EXACT(X)
C
C     COMPUTE ERROR IN CALCULATION
C
      ERROR=YEXACT-YC
C
C     TEST IF DESIRED POINT OF PRINTING IS REACHED
C
      IF((N/ICON*ICON).EQ.N) WRITE(6,1002) X,YP,YC,YEXACT,
     1ERROR
C
C     UPDATE REQUIRED PARAMETER VALUES
C
      Y0=Y1
      Y1=YC
      F1=FP
      X=X+H
 40   CONTINUE
C
C     LOOP BACK TO READ NEXT SET OF INPUT DATA
C
      GO TO 5
 45   STOP
 100  FORMAT(F10.7,I10)
 1000 FORMAT(////41X,'*INPUT DATA SET USED*'//43X,'H    =',
     1F10.7/43X,'ICON =',I10)
 1001 FORMAT(//29X,'*RESULTS OF NYSTROM PREDICTOR-CORRECTOR
      1METHOD *'//10X,'   X   ',5X,'   YP   ',5X,'   YC   ',
     15X,'   YEXACT   ',5X,'   ERROR   '//12X,F10.7,44X,
     1F10.7)
 1002 FORMAT('0',9X,F15.7,2X,F15.7,2X,F15.7,2X,F15.7,2X,
     1F15.7)
      END


      SUBROUTINE RKUTTA(XX,YY,Y1,HH)
      CK1=HH*FCT(XX,YY)
      CK2=HH*FCT(XX+HH/2.0,YY+CK1/2.0)
      CK3=HH*FCT(XX+HH/2.0,YY+CK2/2.0)
      CK4=HH*FCT(XX+HH,YY+CK3)
      Y1=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      XX=XX+HH
      YEXACT=EXACT(XX)
      ERROR=YEXACT-Y1
      WRITE(6,1004) XX,Y1,YEXACT,ERROR
 1004 FORMAT('0',9X,F15.7,2X,F15.7,16X,F15.7,2X,F15.7)
      RETURN
```

```
      END

      FUNCTION FCT(XN,YN)
      FCT=-YN
      RETURN
      END

      FUNCTION EXACT(XK)
      EXACT=EXP(-XK)
      RETURN
      END
```

*INPUT DATA SET USED*

H = 0.5000000

ICON = 2

*RESULTS OF NYSTROM PREDICTOR-CORRECTOR METHOD *

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 0.5000000 | 0.6067709 | | 0.6065306 | -0.0002403 |
| 1.0000000 | 0.3932291 | 0.3636068 | 0.3678794 | 0.0042726 |
| 2.0000000 | 0.1444499 | 0.1298206 | 0.1353353 | 0.0055147 |
| 3.0000000 | 0.0516075 | 0.0463004 | 0.0497871 | 0.0034867 |
| 4.0000000 | 0.0184066 | 0.0165119 | 0.0183156 | 0.0018037 |
| 5.0000000 | 0.0065643 | 0.0058886 | 0.0067379 | 0.0008494 |
| 6.0000000 | 0.0023410 | 0.0021000 | 0.0024788 | 0.0003788 |
| 7.0000000 | 0.0008349 | 0.0007489 | 0.0009119 | 0.0001630 |
| 8.0000000 | 0.0002977 | 0.0002671 | 0.0003355 | 0.0000684 |
| 9.0000000 | 0.0001062 | 0.0000952 | 0.0001234 | 0.0000282 |
| 10.0000000 | 0.0000379 | 0.0000340 | 0.0000454 | 0.0000114 |

*INPUT DATA SET USED*

H = 1.0000000

ICON = 1

*RESULTS OF NYSTROM PREDICTOR-CORRECTOR METHOD *

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 1.0000000 | 0.3750000 | | 0.3678794 | -0.0071206 |
| 2.0000000 | 0.2500000 | 0.1093750 | 0.1353353 | 0.0259603 |
| 3.0000000 | 0.0625000 | 0.0156250 | 0.0497871 | 0.0341621 |
| 4.0000000 | 0.0468750 | -0.0058594 | 0.0183156 | 0.0241750 |
| 5.0000000 | -0.0078125 | -0.0078125 | 0.0067379 | 0.0145504 |
| 6.0000000 | 0.0097656 | -0.0041504 | 0.0024788 | 0.0066291 |
| 7.0000000 | -0.0087891 | -0.0021973 | 0.0009119 | 0.0031091 |
| 8.0000000 | 0.0046387 | -0.0005798 | 0.0003355 | 0.0009153 |
| 9.0000000 | -0.0045166 | -0.0003052 | 0.0001234 | 0.0004286 |
| 10.0000000 | 0.0028381 | 0.0000572 | 0.0000454 | -0.0000118 |

PROGRAM FOUR

HERMITE PREDICTOR-CORRECTOR METHOD. ODE: Y'=-Y WIYH Y(0)=1. TO SOLVE ANOTHER ODE SIMPLY CHANGE INITIAL CONDITIONS AND FUNCTION SUBROUTINES.

```
C       **************************************************************
C       * FORTRAN PROGRAM FOR HERMITE PREDICTOR-CORRECTOR            *
C       * METHOD IN THE NUMERICAL SOLUTION OF THE ORDINARY           *
C       * DIFFERENTIAL EQUATION IN THE FORM OF DY/DX=F(X,Y)          *
C       * WITH THE INITIAL CONDITION Y(X0)=Y0                        *
C       * THE PARAMETERS TO THE PROGRAM HAVE THE FOLLOWING           *
C       * MEANINGS...                                                *
C       *   XMAX_______THE TERMINAL BOUNDARY CONDITION               *
C       *   EPS________THE CONVERGENCE TEST CONSTANT                 *
C       *   MAX________THE MAXIMUM NUMBER OF ITERATIONS              *
C       *   YP_________THE PREDICTED VALUE OF THE NEXT               *
C       *              SOLUTION POINT                                *
C       *   YC_________THE CORRECTED VALUE OF THE NEXT               *
C       *              SOLUTION POINT                                *
C       *   FP_________THE FUNCTION EVALUATION                       *
C       *   H__________THE STEP SIZE TO BE USED TO ADVANCE           *
C       *              THE POINT OF INTEGRATION                      *
C       *   ICON_______THE DESIRED POINT OF PRINTING THE             *
C       *              COMPUTED SOLUTION FOR EVERY FIXED             *
C       *              NUMBER OF INTEGRATION POINTS                  *
C       *   YEXACT_____THE TRUE SOLUTION VALUES                      *
C       *   ERROR______THE ABSOLUTE DEVIATION OF THE NUMERICAL*
C       *              SOLUTION FROM THE TRUE SOLUTION               *
C       *   FCT________FUNCTION SUBROUTINE TO EVALUATE ODE           *
C       *              FUNCTION                                      *
C       *   EXACT______FUNCTION SUBROUTINE TO COMPUTE TRUE           *
C       *              SOLUTION VALUES                               *
C       *   RKUTTA_____THE SUBROUTINE USED TO GENERATE               *
C       *              NEEDED STARTING VALUES                        *
C       **************************************************************
C
C
C
C       INITIALIZE INPUT CONSTANTS
C
        XMAX=10.0
        MAX=2
        EPS=0.000001
C
C       READ STEP SIZE AND CONTROL PRINTING
C
 5      READ(5,100) H,ICON
C
C       SPECIFY INITIAL CONDITION
C
        X0=0.0
        Y0=1.0
        YR=Y0
C
C       TEST FOR END OF INPUT DATA
C
        IF(ICON-0) 45,45,7
C
C       PRINT INPUT DATA SET
C
 7      WRITE(6,1000) H,ICON
        WRITE(6,1001) X0,Y0
C
C       COMPUTE NEXT SOLUTION POINT
C
```

```
      NC=(XMAX+H/2.)/H
      X1=X0+H
      X=X0+2.*H
      CALL RKUTTA(X0,YR,Y1,H)
      F0=FCT(X0,Y0)
      F1=FCT(X1,Y1)
      DO 40 N=2,NC
      M=1
      YP=5.*Y0-4.*Y1+H*(4.*F1+2.*F0)
      FP=FCT(X,YP)
 11   YC=Y0+H/3.*(F0+4.*F1+FP)
      DELY=ABS(YC-YP)
C
C     TEST FOR CONVERGENCE
C
      IF(DELY-EPS) 30,30,15
C
C     TEST IF MAXIMUM NUMBER OF ITERATIONS IS SATISFIED
C
 15   IF(M-MAX) 20,20,30
 20   FP=FCT(X,YC)
      M=M+1
      GO TO 11
C
C     COMPUTE TRUE SOLUTION VALUE
C
 30   YEXACT=EXACT(X)
C
C     COMPUTE ERROR IN CALCULATION
C
      ERROR=YEXACT-YC
C
C     TEST IF DESIRED POINT OF PRINTING IS REACHED
C
      IF((N/ICON*ICON).EQ.N) WRITE(6,1002) X,YP,YC,YEXACT,
     1ERROR
C
C     UPDATE REQUIRED PARAMETER VALUES
C
      Y0=Y1
      Y1=YC
      F0=F1
      F1=FP
      X=X+H
 40   CONTINUE
C
C     LOOP BACK TO READ NEXT SET OF INPUT DATA
C
      GO TO 5
 45   STOP
 100  FORMAT(F10.7,I10)
 1000 FORMAT(////41X,'*INPUT DATA SET USED*'//43X,'H    =',
     1F10.7/43X,'ICON =',I10)
 1001 FORMAT(//29X,'*RESULTS OF HERMITE PREDICTOR-CORRECTOR
     1METHOD *'//10X,'    X   ',5X,'    YP    ',5X,'    YC    ',
     15X,'   YEXACT    ',5X,'    ERROR    '//12X,F10.7,44X,
     1F10.7)
 1002 FORMAT('0',9X,F15.7,2X,F15.7,2X,F15.7,2X,F15.7,2X,
     1F15.7)
      END


      SUBROUTINE RKUTTA(XX,YY,Y1,HH)
      CK1=HH*FCT(XX,YY)
      CK2=HH*FCT(XX+HH/2.0,YY+CK1/2.0)
      CK3=HH*FCT(XX+HH/2.0,YY+CK2/2.0)
      CK4=HH*FCT(XX+HH,YY+CK3)
      Y1=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      XX=XX+HH
      YEXACT=EXACT(XX)
      ERROR=YEXACT-Y1
      WRITE(6,1004) XX,Y1,YEXACT,ERROR
```

```
1004 FORMAT('0',9X,F15.7,2X,F15.7,16X,F15.7,2X,F15.7)
     RETURN
     END


     FUNCTION FCT(XN,YN)
     FCT=-YN
     RETURN
     END


     FUNCTION EXACT(XK)
     EXACT=EXP(-XK)
     RETURN
     END
```

*INPUT DATA SET USED*

H = 0.5000000

ICON = 2

*RESULTS OF HERMITE PREDICTOR-CORRECTOR METHOD *

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 0.5000000 | 0.6067709 | | 0.6065306 | -0.0002403 |
| 1.0000000 | 0.3593760 | 0.3675976 | 0.3678794 | 0.0002818 |
| 2.0000000 | 0.1296880 | 0.1349390 | 0.1353353 | 0.0003963 |
| 3.0000000 | 0.0439692 | 0.0491728 | 0.0497871 | 0.0006143 |
| 4.0000000 | 0.0106042 | 0.0173813 | 0.0183156 | 0.0009343 |
| 5.0000000 | -0.0043826 | 0.0053371 | 0.0067379 | 0.0014008 |
| 6.0000000 | -0.0139239 | 0.0003908 | 0.0024788 | 0.0020879 |
| 7.0000000 | -0.0234190 | -0.0021938 | 0.0009119 | 0.0031057 |
| 8.0000000 | -0.0358057 | -0.0042811 | 0.0003355 | 0.0046165 |
| 9.0000000 | -0.0535792 | -0.0067376 | 0.0001234 | 0.0068610 |
| 10.0000000 | -0.0797584 | -0.0101509 | 0.0000454 | 0.0101963 |

*INPUT DATA SET USED*

H = 1.0000000

ICON = 1

*RESULTS OF HERMITE PREDICTOR-CORRECTOR METHOD *

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 1.0000000 | 0.3750000 | | 0.3678794 | -0.0071206 |
| 2.0000000 | 0.0000000 | 0.1296299 | 0.1353353 | 0.0057054 |
| 3.0000000 | 0.1620350 | 0.0732166 | 0.0497871 | -0.0234295 |
| 4.0000000 | -0.2105594 | -0.0092715 | 0.0183156 | 0.0275871 |
| 5.0000000 | 0.3834665 | 0.0599074 | 0.0067379 | -0.0531695 |
| 6.0000000 | 0.6344536 | -0.0839149 | 0.0024788 | 0.0863936 |
| 7.0000000 | 1.0731880 | 0.1479157 | 0.0009119 | -0.1470038 |
| 8.0000000 | -1.8065000 | -0.2466852 | 0.0003355 | 0.2470207 |
| 9.0000000 | 3.0441850 | 0.4165850 | 0.0001234 | -0.4164615 |
| 10.0000000 | -5.1285590 | -0.7014816 | 0.0000454 | 0.7015270 |

PROGRAM FIVE

HAMMING PREDICTOR-CORRECTOR METHOD. ODE: Y'=-Y WIYH Y(0)=1. TO SOLVE ANOTHER ODE SIMPLY CHANGE INITIAL CONDITIONS AND FUNCTION SUBROUTINES.

```
C       ************************************************************
C       * FORTRAN PROGRAM FOR HAMMING PREDICTOR-CORRECTOR          *
C       * METHOD IN THE NUMERICAL SOLUTION OF THE ORDINARY         *
C       * DIFFERENTIAL EQUATION IN THE FORM OF DY/DX=F(X,Y)        *
C       * WITH THE INITIAL CONDITION Y(XO)=YO                      *
C       * THE PARAMETERS TO THE PROGRAM HAVE THE FOLLOWING         *
C       * MEANINGS...                                              *
C       *   XMAX______THE TERMINAL BOUNDARY CONDITION              *
C       *   EPS_______THE CONVERGENCE TEST CONSTANT                *
C       *   MAX_______THE MAXIMUM NUMBER OF ITERATIONS             *
C       *   YPM_______PREDICTED SOLUTION VALUE                     *
C       *   YP________MODIFIED PREDICTED SOLUTION VALUE            *
C       *   YCM_______CORRECTED SOLUTION VALUE                     *
C       *   YC________MODIFIED CORRECTED SOLUTION VALUE            *
C       *   FP________THE FUNCTION EVALUATION                      *
C       *   H_________THE STEP SIZE TO BE USED TO ADVANCE          *
C       *             THE POINT OF INTEGRATION                     *
C       *   ICON______THE DESIRED POINT OF PRINTING THE            *
C       *             COMPUTED SOLUTION FOR EVERY FIXED            *
C       *             NUMBER OF INTEGRATION POINTS                 *
C       *   YEXACT____THE TRUE SOLUTION VALUES                     *
C       *   ERROR_____THE ABSOLUTE DEVIATION OF THE NUMERICAL*
C       *             SOLUTION FROM THE TRUE SOLUTION              *
C       *   FCT_______FUNCTION SUBROUTINE TO EVALUATE ODE          *
C       *             FUNCTION                                     *
C       *   EXACT_____FUNCTION SUBROUTINE TO COMPUTE TRUE          *
C       *             SOLUTION VALUES                              *
C       *   RKUTTA____THE SUBROUTINE USED TO GENERATE              *
C       *             NEEDED STARTING VALUES                       *
C       ************************************************************
C
C
C       INITIALIZE INPUT CONSTANTS
C
        XMAX=10.0
        MAX=2
        EPS=0.000001
        K=3
C
C       READ STEP SIZE AND CONTROL PRINTING
C
 5      READ(5,100) H,ICON
C
C       SPECIFY INITIAL CONDITION
C
        ET=0.0
        XO=0.0
        YO=1.0
        YR=YO
C
C       TEST FOR END OF INPUT DATA
C
        IF(ICON-0) 45,45,7
C
C       PRINT INPUT DATA SET
C
 7      WRITE(6,1000) H,ICON
        WRITE(6,1001) XO,YO
C
```

```
C       COMPUTE NEXT SOLUTION POINT
C
        NC=(XMAX+H/2.)/H
        X1=X0+H
        X2=X0+2.*H
        X3=X0+3.*H
        X=X0+4.*H
        CALL RKUTTA(X0,YR,Y1,Y2,Y3,K,H)
        F1=FCT(X1,Y1)
        F2=FCT(X2,Y2)
        F3=FCT(X3,Y3)
        DO 40 N=4,NC
        M=1
        YPM=Y0+4.*H/3.*(2.*F1-F2+2.*F3)
        YP=YPM-112./121.*ET
        FP=FCT(X,YP)
 11     YCM=1./8.*(9.*Y3-Y1)+3.*H/8.*(FP+2.*F3-F2)
        ET=YPM-YCM
        YC=YCM+9./121.*ET
        DELY=ABS(YC-YP)
C
C       TEST FOR CONVERGENCE
C
        IF(DELY-EPS) 30,30,15
C
C       TEST IF MAXIMUM NUMBER OF ITERATIONS IS SATISFIED
C
 15     IF(M-MAX) 20,20,30
 20     FP=FCT(X,YC)
        M=M+1
        GO TO 11
C
C       COMPUTE TRUE SOLUTION VALUE
C
 30     YEXACT=EXACT(X)
C
C       COMPUTE ERROR IN CALCULATION
C
        ERROR=YEXACT-YC
C
C       TEST IF DESIRED POINT OF PRINTING IS REACHED
C
        IF((N/ICON*ICON).EQ.N) WRITE(6,1002) X,YP,YC,YEXACT,
       1ERROR
C
C       UPDATE REQUIRED PARAMETER VALUES
C
        F1=F2
        F2=F3
        F3=FP
        Y0=Y1
        Y1=Y2
        Y2=Y3
        Y3=YC
        X=X+H
 40     CONTINUE
C
C       LOOP BACK TO READ NEXT SET OF INPUT DATA
C
        GO TO 5
 45     STOP
 100    FORMAT(F10.7,I10)
 1000   FORMAT(////41X,'*INPUT DATA SET USED*'//43X,'H    =',
       1F10.7/43X,'ICON =',I10)
 1001   FORMAT(//29X,'*RESULTS OF HAMMING PREDICTOR-CORRECTOR
       1METHOD *'//10X,'    X   ',5X,'    YP    ',5X,'    YC    ',
       15X,'    YEXACT    ',5X,'    ERROR    '//12X,F10.7,44X,
       1F10.7)
 1002   FORMAT('0',9X,F15.7,2X,F15.7,2X,F15.7,2X,F15.7,2X,
       1F15.7)
        END
```

```
      SUBROUTINE RKUTTA(XX,YY,Y1,Y2,Y3,KK,HH)
      NN=1
17    CK1=HH*FCT(XX,YY)
      CK2=HH*FCT(XX+HH/2.0,YY+CK1/2.0)
      CK3=HH*FCT(XX+HH/2.0,YY+CK2/2.0)
      CK4=HH*FCT(XX+HH,YY+CK3)
      GO TO (101,102,103),NN
101   Y1=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y1
77    XX=XX+HH
      YEXACT=EXACT(XX)
      ERROR=YEXACT-YY
      WRITE(6,1004) XX,YY,YEXACT,ERROR
      NN=NN+1
      IF(NN-KK) 17,17,99
102   Y2=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y2
      GO TO 77
103   Y3=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y3
      GO TO 77
1004  FORMAT('0',9X,F15.7,2X,F15.7,16X,F15.7,2X,F15.7)
99    RETURN
      END

      FUNCTION FCT(XN,YN)
      FCT=-YN
      RETURN
      END

      FUNCTION EXACT(XK)
      EXACT=EXP(-XK)
      RETURN
      END
```

*INPUT DATA SET USED*

H = 0.5000000

ICON = 2

*RESULTS OF HAMMING PREDICTOR-CORRECTOR METHOD *

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 0.5000000 | 0.6067709 | | 0.6065306 | -0.0002403 |
| 1.0000000 | 0.3681710 | | 0.3678794 | -0.0002916 |
| 1.5000000 | 0.2233955 | | 0.2231301 | -0.0002654 |
| 2.0000000 | 0.1385594 | 0.1355409 | 0.1353353 | -0.0002056 |
| 3.0000000 | 0.0494538 | 0.0499289 | 0.0497871 | -0.0001418 |
| 4.0000000 | 0.0184383 | 0.0183963 | 0.0183156 | -0.0000806 |
| 5.0000000 | 0.0068499 | 0.0067794 | 0.0067379 | -0.0000415 |
| 6.0000000 | 0.0025701 | 0.0024990 | 0.0024788 | -0.0000202 |
| 7.0000000 | 0.0009748 | 0.0009215 | 0.0009119 | -0.0000096 |
| 8.0000000 | 0.0003766 | 0.0003401 | 0.0003355 | -0.0000046 |
| 9.0000000 | 0.0001496 | 0.0001256 | 0.0001234 | -0.0000022 |
| 10.0000000 | 0.0000619 | 0.0000465 | 0.0000454 | -0.0000011 |

*INPUT DATA SET USED*

H = 1.0000000

ICON = 1

*RESULTS OF HAMMING PREDICTOR-CORRECTOR METHOD *

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 1.0000000 | 0.3750000 | | 0.3678794 | -0.0071206 |
| 2.0000000 | 0.1406250 | | 0.1353353 | -0.0052897 |
| 3.0000000 | 0.0527344 | | 0.0497871 | -0.0029473 |
| 4.0000000 | 0.0468752 | 0.0190868 | 0.0183156 | -0.0007712 |
| 5.0000000 | -0.0199183 | 0.0056581 | 0.0067379 | 0.0010798 |
| 6.0000000 | 0.0245465 | 0.0057380 | 0.0024788 | -0.0032592 |
| 7.0000000 | -0.0516121 | -0.0008928 | 0.0009119 | 0.0018047 |
| 8.0000000 | 0.0793315 | 0.0053606 | 0.0003355 | -0.0050251 |
| 9.0000000 | -0.1184798 | -0.0069360 | 0.0001234 | 0.0070594 |
| 10.0000000 | 0.1838089 | 0.0110968 | 0.0000454 | -0.0110514 |

PROGRAM SIX

SECOND ORDER ADAMS PREDICTOR-CORRECTOR METHOD. ODE: Y'=-Y WITH Y(0)=1. TO SOLVE ANOTHER ODE CHANGE INITIAL CONDITIONS AND FUNCTION SUBROUTINES.

```
C       **************************************************************
C       * FORTRAN PROGRAM FOR SECOND ORDER ADAMS PREDICTOR-           *
C       * CORRECTOR METHOD IN THE NUMERICAL SOLUTION OF THE           *
C       * ORDINARY DIFFERENTIAL EQUATION IN THE FORM OF               *
C       * DY/DX=F(X,Y) WITH THE INITIAL CONDITION Y(XO)=YO            *
C       * THE PARAMETERS TO THE PROGRAM HAVE THE FOLLOWING            *
C       * MEANINGS...                                                 *
C       *   XMAX_______THE TERMINAL BOUNDARY CONDITION                *
C       *   EPS________THE CONVERGENCE TEST CONSTANT                  *
C       *   MAX________THE MAXIMUM NUMBER OF ITERATIONS               *
C       *   YP_________THE PREDICTED VALUE OF THE NEXT                *
C       *              SOLUTION POINT                                 *
C       *   YC_________THE CORRECTED VALUE OF THE NEXT                *
C       *              SOLUTION POINT                                 *
C       *   FP_________THE FUNCTION EVALUATION                        *
C       *   H__________THE STEP SIZE TO BE USED TO ADVANCE            *
C       *              THE POINT OF INTEGRATION                       *
C       *   ICON_______THE DESIRED POINT OF PRINTING THE              *
C       *              COMPUTED SOLUTION FOR EVERY FIXED              *
C       *              NUMBER OF INTEGRATION POINTS                   *
C       *   YEXACT_____THE TRUE SOLUTION VALUES                       *
C       *   ERROR______THE ABSOLUTE DEVIATION OF THE NUMERICAL*
C       *              SOLUTION FROM THE TRUE SOLUTION                *
C       *   FCT________FUNCTION SUBROUTINE TO EVALUATE ODE            *
C       *              FUNCTION.                                      *
C       *   EXACT______FUNCTION SUBROUTINE TO COMPUTE TRUE            *
C       *              SOLUTION VALUES                                *
C       *   RKUTTA_____THE SUBROUTINE USED TO GENERATE                *
C       *              NEEDED STARTING VALUES                         *
C       **************************************************************
C
C
C       INITIALIZE INPUT CONSTANTS
C
        XMAX=10.0
        MAX=2
        EPS=0.000001
C
C       READ STEP SIZE AND CONTROL PRINTING
C
 5      READ(5,100) H,ICON
C
C       SPECIFY INITIAL CONDITION
C
        XO=0.0
        YO=1.0
        YR=YO
C
C       TEST FOR END OF INPUT DATA
C
        IF(ICON-0) 45,45,7
C
C       PRINT INPUT DATA SET
C
 7      WRITE(6,1000) H,ICON
        WRITE(6,1001) XO,YO
C
C       COMPUTE NEXT SOLUTION POINT
C
```

```
      NC=(XMAX+H/2.)/H
      F0=FCT(X0,Y0)
      X1=X0+H
      X=X0+2.*H
      CALL RKUTTA(X0,YR,Y1,H)
      F1=FCT(X1,Y1)
      DO 40 N=2,NC
      M=1
      YP=Y1+H/2.*(3.*F1-F0)
      FP=FCT(X,YP)
 11   YC=Y1+H/2.*(FP+F1)
      DELY=ABS(YC-YP)
C
C     TEST FOR CONVERGENCE
C
      IF(DELY-EPS) 30,30,15
C
C     TEST IF MAXIMUM NUMBER OF ITERATIONS IS SATISFIED
C
 15   IF(M-MAX) 20,20,30
 20   FP=FCT(X,YC)
      M=M+1
      GO TO 11
C
C     COMPUTE TRUE SOLUTION VALUE
C
 30   YEXACT=EXACT(X)
C
C     COMPUTE ERROR IN CALCULATION
C
      ERROR=YEXACT-YC
C
C     TEST IF DESIRED POINT OF PRINTING IS REACHED
C
      IF((N/ICON*ICON).EQ.N) WRITE(6,1002) X,YP,YC,YEXACT,
     1ERROR
C
C     UPDATE REQUIRED PARAMETER VALUES
C
      Y0=Y1
      Y1=YC
      F1=FP
      X=X+H
 40   CONTINUE
C
C     LOOP BACK TO READ NEXT SET OF INPUT DATA
C
      GO TO 5
 45   STOP
 100  FORMAT(F10.7,I10)
 1000 FORMAT(////41X,'*INPUT DATA SET USED*'//43X,'H    =',
     1F10.7/43X,'ICON =',I10)
 1001 FORMAT(//29X,'**RESULTS OF ADAM2 PREDICTOR-CORRECTOR
     1METHOD**'//10X,'    X   ',5X,'    YP    ',5X,'    YC    ',
     15X,'   YEXACT    ',5X,'    ERROR    '//12X,F10.7,44X,
     1F10.7)
 1002 FORMAT('0',9X,F15.7,2X,F15.7,2X,F15.7,2X,F15.7,2X,
     1F15.7)
      END


      SUBROUTINE RKUTTA(XX,YY,Y1,HH)
      CK1=HH*FCT(XX,YY)
      CK2=HH*FCT(XX+HH/2.0,YY+CK1/2.0)
      CK3=HH*FCT(XX+HH/2.0,YY+CK2/2.0)
      CK4=HH*FCT(XX+HH,YY+CK3)
      Y1=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      XX=XX+HH
      YEXACT=EXACT(XX)
      ERROR=YEXACT-Y1
      WRITE(6,1004) XX,Y1,YEXACT,ERROR
 1004 FORMAT('0',9X,F15.7,2X,F15.7,16X,F15.7,2X,F15.7)
```

```
      RETURN
      END

      FUNCTION FCT(XN,YN)
      FCT=-YN
      RETURN
      END

      FUNCTION EXACT(XK)
      EXACT=EXP(-XK)
      RETURN
      END
```

*INPUT DATA SET USED*

H = 0.5000000

ICON = 2

**RESULTS OF ADAM2 PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 0.5000000 | 0.6067709 | | 0.6065306 | -0.0002403 |
| 1.0000000 | 0.4016927 | 0.3634746 | 0.3678794 | 0.0044048 |
| 2.0000000 | 0.2968013 | 0.1248231 | 0.1353353 | 0.0105122 |
| 3.0000000 | 0.2556222 | 0.0349784 | 0.0497871 | 0.0148087 |
| 4.0000000 | 0.2401217 | 0.0011687 | 0.0183156 | 0.0171469 |
| 5.0000000 | 0.2342887 | -0.0115542 | 0.0067379 | 0.0182921 |
| 6.0000000 | 0.2320936 | -0.0163420 | 0.0024788 | 0.0188207 |
| 7.0000000 | 0.2312676 | -0.0181437 | 0.0009119 | 0.0190556 |
| 8.0000000 | 0.2309567 | -0.0188217 | 0.0003355 | 0.0191571 |
| 9.0000000 | 0.2308398 | -0.0190768 | 0.0001234 | 0.0192002 |
| 10.0000000 | 0.2307958 | -0.0191728 | 0.0000454 | 0.0192182 |

*INPUT DATA SET USED*

H = 1.0000000

ICON = 1

**RESULTS OF ADAM2 PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 1.0000000 | 0.3750000 | | 0.3678794 | -0.0071206 |
| 2.0000000 | 0.3125000 | 0.1015625 | 0.1353353 | 0.0337728 |
| 3.0000000 | 0.3437500 | -0.0312500 | 0.0497871 | 0.0810370 |
| 4.0000000 | 0.3281250 | -0.0996094 | 0.0183156 | 0.1179250 |
| 5.0000000 | 0.3359375 | -0.1328125 | 0.0067379 | 0.1395504 |
| 6.0000000 | 0.3320313 | -0.1499023 | 0.0024788 | 0.1523811 |
| 7.0000000 | 0.3339844 | -0.1582031 | 0.0009119 | 0.1591150 |
| 8.0000000 | 0.3330078 | -0.1624756 | 0.0003355 | 0.1628110 |
| 9.0000000 | 0.3334961 | -0.1645508 | 0.0001234 | 0.1646742 |
| 10.0000000 | 0.3332520 | -0.1656189 | 0.0000454 | 0.1656643 |

PROGRAM SEVEN

THIRD ORDER ADAMS PREDICTOR-CORRECTOR METHOD. ODE: Y'=-Y WITH Y(0)=1. TO SOLVE ANOTHER ODE CHANGE INITIAL CONDITIONS AND FUNCTION SUBROUTINES.

```
C        ***************************************************************
C        * FORTRAN PROGRAM FOR THIRD ORDER ADAMS PREDICTOR-            *
C        * CORRECTOR METHOD IN THE NUMERICAL SOLUTION OF THE           *
C        * ORDINARY DIFFERENTIAL EQUATION IN THE FORM OF               *
C        * DY/DX=F(X,Y) WITH THE INITIAL CONDITION Y(X0)=Y0            *
C        * THE PARAMETERS TO THE PROGRAM HAVE THE FOLLOWING            *
C        * MEANINGS...                                                 *
C        *    XMAX_______THE TERMINAL BOUNDARY CONDITION               *
C        *    EPS________THE CONVERGENCE TEST CONSTANT                 *
C        *    MAX________THE MAXIMUM NUMBER OF ITERATIONS              *
C        *    YP_________THE PREDICTED VALUE OF THE NEXT               *
C        *               SOLUTION POINT                                *
C        *    YC_________THE CORRECTED VALUE OF THE NEXT               *
C        *               SOLUTION POINT                                *
C        *    FP_________THE FUNCTION EVALUATION                       *
C        *    H__________THE STEP SIZE TO BE USED TO ADVANCE           *
C        *               THE POINT OF INTEGRATION                      *
C        *    ICON_______THE DESIRED POINT OF PRINTING THE             *
C        *               COMPUTED SOLUTION FOR EVERY FIXED             *
C        *               NUMBER OF INTEGRATION POINTS                  *
C        *    YEXACT_____THE TRUE SOLUTION VALUES                      *
C        *    ERROR______THE ABSOLUTE DEVIATION OF THE NUMERICAL*
C        *               SOLUTION FROM THE TRUE SOLUTION               *
C        *    FCT________FUNCTION SUBROUTINE TO EVALUATE ODE           *
C        *               FUNCTION                                      *
C        *    EXACT______FUNCTION SUBROUTINE TO COMPUTE TRUE           *
C        *               SOLUTION VALUES                               *
C        *    RKUTTA_____THE SUBROUTINE USED TO GENERATE               *
C        *               NEEDED STARTING VALUES                        *
C        ***************************************************************
C
C
C        INITIALIZE INPUT CONSTANTS
C
         XMAX=10.0
         MAX=2
         EPS=0.000001
         K=2
C
C        READ STEP SIZE AND CONTROL PRINTING
C
   5     READ(5,100) H,ICON
C
C        SPECIFY INITIAL CONDITION
         X0=0.0
         Y0=1.0
         YR=Y0
C
C        TEST FOR END OF INPUT DATA
C
         IF(ICON-0) 45,45,7
C
C        PRINT INPUT DATA SET
C
   7     WRITE(6,1000) H,ICON
         WRITE(6,1001) X0,Y0
C
C        COMPUTE NEXT SOLUTION POINT
C
```

```
      NC=(XMAX+H/2.)/H
      F0=FCT(X0,Y0)
      X1=X0+H
      X2=X0+2.*H
      X=X0+3.*H
      CALL RKUTTA(X0,YR,Y1,Y2,K,H)
      F1=FCT(X1,Y1)
      F2=FCT(X2,Y2)
      DO 40 N=3,NC
      M=1
      YP=Y1+H/12.*(23.*F2-16.*F1+5.*F0)
      FP=FCT(X,YP)
 11   YC=Y2+H/12.*(5.*FP+8.*F2-F1)
      DELY=ABS(YC-YP)
C
C     TEST FOR CONVERGENCE
C
      IF(DELY-EPS) 30,30,15
C
C     TEST IF MAXIMUM NUMBER OF ITERATIONS IS SATISFIED
C
 15   IF(M-MAX) 20,20,30
 20   FP=FCT(X,YC)
      M=M+1
      GO TO 11
C
C     COMPUTE TRUE SOLUTION VALUE
C
 30   YEXACT=EXACT(X)
C
C     COMPUTE ERROR IN CALCULATION
C
      ERROR=YEXACT-YC
C
C     TEST IF DESIRED POINT OF PRINTING IS REACHED
C
      IF((N/ICON*ICON).EQ.N) WRITE(6,1002) X,YP,YC,YEXACT,
     1ERROR
C
C     UPDATE REQUIRED PARAMETER VALUES
C
      F0=F1
      F1=F2
      F2=FP
      Y0=Y1
      Y1=Y2
      Y2=YC
      X=X+H
 40   CONTINUE
C
C     LOOP BACK TO READ NEXT SET OF INPUT DATA
C
      GO TO 5
 45   STOP
 100  FORMAT(F10.7,I10)
 1000 FORMAT(////41X,'*INPUT DATA SET USED*'//43X,'H    =',
     1F10.7/43X,'ICON =',I10)
 1001 FORMAT(//29X,'**RESULTS OF ADAM3 PREDICTOR-CORRECTOR
     1METHOD**'//10X,'    X    ',5X,'    YP    ',5X,'    YC    ',
     15X,'   YEXACT    ',5X,'   ERROR    '//12X,F10.7,44X,
     1F10.7)
 1002 FORMAT('0',9X,F15.7,2X,F15.7,2X,F15.7,2X,F15.7,2X,
     1F15.7)
      END


      SUBROUTINE RKUTTA(XX,YY,Y1,Y2,KK,HH)
      NN=1
 17   CK1=HH*FCT(XX,YY)
      CK2=HH*FCT(XX+HH/2.0,YY+CK1/2.0)
      CK3=HH*FCT(XX+HH/2.0,YY+CK2/2.0)
      CK4=HH*FCT(XX+HH,YY+CK3)
```

```
      GO TO (101,102),NN
101   Y1=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y1
77    XX=XX+HH
      YEXACT=EXACT(XX)
      ERROR=YEXACT-YY
      WRITE(6,1004) XX,YY,YEXACT,ERROR
      NN=NN+1
      IF(NN-KK) 17,17,99
102   Y2=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y2
      GO TO 77
1004  FORMAT('0',9X,F15.7,2X,F15.7,16X,F15.7,2X,F15.7)
99    RETURN
      END

      FUNCTION FCT(XN,YN)
      FCT=-YN
      RETURN
      END

      FUNCTION EXACT(XK)
      EXACT=EXP(-XK)
      RETURN
      END
```

*INPUT DATA SET USED*

H = 0.5000000

ICON = 2

**RESULTS OF ADAM3 PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 0.5000000 | 0.6067709 | | 0.6065306 | -0.0002403 |
| 1.0000000 | 0.3681710 | | 0.3678794 | -0.0002916 |
| 2.0000000 | 0.2630881 | 0.1307259 | 0.1353353 | 0.0046093 |
| 3.0000000 | 0.0949250 | 0.0457278 | 0.0497871 | 0.0040593 |
| 4.0000000 | 0.0333163 | 0.0160214 | 0.0183156 | 0.0022942 |
| 6.0000000 | 0.0041118 | 0.0019661 | 0.0024788 | 0.0005126 |
| 7.0000000 | 0.0014410 | 0.0006887 | 0.0009119 | 0.0002232 |
| 8.0000000 | 0.0005048 | 0.0002412 | 0.0003355 | 0.0000942 |
| 9.0000000 | 0.0001768 | 0.0000845 | 0.0001234 | 0.0000389 |
| 10.0000000 | 0.0000619 | 0.0000296 | 0.0000454 | 0.0000158 |

*INPUT DATA SET USED*

H = 1.0000000

ICON = 1

**RESULTS OF ADAM3 PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 1.0000000 | 0.3750000 | | 0.3678794 | -0.0071206 |
| 2.0000000 | 0.1406250 | | 0.1353353 | -0.0052897 |
| 3.0000000 | 0.1888022 | 0.0454788 | 0.0497871 | 0.0043082 |
| 4.0000000 | 0.0217022 | 0.0021872 | 0.0183156 | 0.0161284 |
| 5.0000000 | 0.0785821 | -0.0024490 | 0.0067379 | 0.0091869 |
| 6.0000000 | -0.0525024 | -0.0057783 | 0.0024788 | 0.0082570 |
| 7.0000000 | -0.0479046 | 0.0015025 | 0.0009119 | -0.0005906 |
| 8.0000000 | -0.0577729 | -0.0018528 | 0.0003355 | 0.0021883 |
| 9.0000000 | 0.0527026 | 0.0029584 | 0.0001234 | -0.0028350 |
| 10.0000000 | -0.0540226 | -0.0020290 | 0.0000454 | 0.0020744 |

PROGRAM EIGHT

FOURTH ORDER ADAMS PREDICTOR-CORRECTOR METHOD. ODE: Y'=-Y WITH Y(0)=1. TO SOLVE ANOTHER ODE CHANGE INITIAL CONDITIONS AND FUNCTION SUBROUTINES.

```
C       **********************************************************
C       * FORTRAN PROGRAM FOR FOURTH ORDER ADAMS PREDICTOR-     *
C       * CORRECTOR METHOD IN THE NUMERICAL SOLUTION OF THE     *
C       * ORDINARY DIFFERENTIAL EQUATION IN THE FORM OF         *
C       * DY/DX=F(X,Y) WITH THE INITIAL CONDITION Y(XO)=YO      *
C       * THE PARAMETERS TO THE PROGRAM HAVE THE FOLLOWING      *
C       * MEANINGS...                                           *
C       *   XMAX______THE TERMINAL BOUNDARY CONDITION           *
C       *   EPS_______THE CONVERGENCE TEST CONSTANT             *
C       *   MAX_______THE MAXIMUM NUMBER OF ITERATIONS          *
C       *   YP________THE PREDICTED VALUE OF THE NEXT           *
C       *             SOLUTION POINT                            *
C       *   YC________THE CORRECTED VALUE OF THE NEXT           *
C       *             SOLUTION POINT                            *
C       *   FP________THE FUNCTION EVALUATION                   *
C       *   H_________THE STEP SIZE TO BE USED TO ADVANCE       *
C       *             THE POINT OF INTEGRATION                  *
C       *   ICON______THE DESIRED POINT OF PRINTING THE         *
C       *             COMPUTED SOLUTION FOR EVERY FIXED         *
C       *             NUMBER OF INTEGRATION POINTS              *
C       *   YEXACT____THE TRUE SOLUTION VALUES                  *
C       *   ERROR_____THE ABSOLUTE DEVIATION OF THE NUMERICAL*
C       *             SOLUTION FROM THE TRUE SOLUTION           *
C       *   FCT_______FUNCTION SUBROUTINE TO EVALUATE ODE       *
C       *             FUNCTION                                  *
C       *   EXACT_____FUNCTION SUBROUTINE TO COMPUTE TRUE       *
C       *             SOLUTION VALUES                           *
C       *   RKUTTA____THE SUBROUTINE USED TO GENERATE           *
C       *             NEEDED STARTING VALUES                    *
C       **********************************************************
C
C
C       INITIALIZE INPUT CONSTANTS
C
        XMAX=10.0
        MAX=2
        EPS=0.000001
        K=3
C
C       READ STEP SIZE AND CONTROL PRINTING
C
  5     READ(5,100) H,ICON
C
C       SPECIFY INITIAL CONDITION
C
        X0=0.0
        Y0=1.0
        YR=Y0
C
C       TEST FOR END OF INPUT DATA
C
        IF(ICON-0) 45,45,7
C
C       PRINT INPUT DATA SET
C
  7     WRITE(6,1000) H,ICON
        WRITE(6,1001) X0,Y0
C
C       COMPUTE NEXT SOLUTION POINT
```

```
C
      NC=(XMAX+H/2.)/H
      F0=FCT(X0,Y0)
      X1=X0+H
      X2=X0+2.*H
      X3=X0+3.*H
      X=X0+4.*H
      CALL RKUTTA(X0,YR,Y1,Y2,Y3,K,H)
      F1=FCT(X1,Y1)
      F2=FCT(X2,Y2)
      F3=FCT(X3,Y3)
      DO 40 N=4,NC
      M=1
      YP=Y3+H/24.*(55.*F3-59.*F2+37.*F1-9.*F0)
      FP=FCT(X,YP)
 11   YC=Y3+H/24.*(9.*FP+19.*F3-5.*F2+F1)
      DELY=ABS(YC-YP)
C
C     TEST FOR CONVERGENCE
C
      IF(DELY-EPS) 30,30,15
C
C     TEST IF MAXIMUM NUMBER OF ITERATIONS IS SATISFIED
C
 15   IF(M-MAX) 20,20,30
 20   FP=FCT(X,YC)
      M=M+1
      GO TO 11
C
C     COMPUTE TRUE SOLUTION VALUE
C
 30   YEXACT=EXACT(X)
C
C     COMPUTE ERROR IN CALCULATION
C
      ERROR=YEXACT-YC
C
C     TEST IF DESIRED POINT OF PRINTING IS REACHED
C
      IF((N/ICON*ICON).EQ.N) WRITE(6,1002) X,YP,YC,YEXACT,
     1ERROR
C
C     UPDATE REQUIRED PARAMETER VALUES
C
      Y0=Y1
      Y1=Y2
      Y2=Y3
      Y3=YC
      F0=F1
      F1=F2
      F2=F3
      F3=FP
      X=X+H
 40   CONTINUE
C
C     LOOP BACK TO READ NEXT SET OF INPUT DATA
C
      GO TO 5
 45   STOP
 100  FORMAT(F10.7,I10)
 1000 FORMAT(////41X,'*INPUT DATA SET USED*'//43X,'H    =',
     1F10.7/43X,'ICON =',I10)
 1001 FORMAT(//29X,'**RESULTS OF ADAM4 PREDICTOR-CORRECTOR
      1METHOD**'//10X,'   X   ',5X,'   YP    ',5X,'   YC    ',
     15X,'   YEXACT   ',5X,'   ERROR   '//12X,F10.7,44X,
     1F10.7)
 1002 FORMAT('0',9X,F15.7,2X,F15.7,2X,F15.7,2X,F15.7,2X,
     1F15.7)
      END
```

```
      SUBROUTINE RKUTTA(XX,YY,Y1,Y2,Y3,KK,HH)
      NN=1
17    CK1=HH*FCT(XX,YY)
      CK2=HH*FCT(XX+HH/2.0,YY+CK1/2.0)
      CK3=HH*FCT(XX+HH/2.0,YY+CK2/2.0)
      CK4=HH*FCT(XX+HH,YY+CK3)
      GO TO (101,102,103),NN
101   Y1=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y1
77    XX=XX+HH
      YEXACT=EXACT(XX)
      ERROR=YEXACT-YY
      WRITE(6,1004) XX,YY,YEXACT,ERROR
      NN=NN+1
      IF(NN-KK) 17,17,99
102   Y2=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y2
      GO TO 77
103   Y3=YY+(CK1+2.0*CK2+2.0*CK3+CK4)/6.0
      YY=Y3
      GO TO 77
1004  FORMAT('0',9X,F15.7,2X,F15.7,16X,F15.7,2X,F15.7)
99    RETURN
      END


      FUNCTION FCT(XN,YN)
      FCT=-YN
      RETURN
      END


      FUNCTION EXACT(XK)
      EXACT=EXP(-XK)
      RETURN
      END
```

*INPUT DATA SET USED*

H = 0.5000000

ICON = 2

**RESULTS OF ADAM4 PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 0.5000000 | 0.6067709 | | 0.6065306 | -0.0002403 |
| 1.0000000 | 0.3681710 | | 0.3678794 | -0.0002916 |
| 1.5000000 | 0.2233955 | | 0.2231301 | -0.0002654 |
| 2.0000000 | 0.1397457 | 0.1352788 | 0.1353353 | 0.0000565 |
| 3.0000000 | 0.0512678 | 0.0495746 | 0.0497871 | 0.0002124 |
| 4.0000000 | 0.0187947 | 0.0181669 | 0.0183156 | 0.0001488 |
| 5.0000000 | 0.0068879 | 0.0066569 | 0.0067379 | 0.0000810 |
| 6.0000000 | 0.0025232 | 0.0024393 | 0.0024788 | 0.0000394 |
| 7.0000000 | 0.0009245 | 0.0008938 | 0.0009119 | 0.0000180 |
| 8.0000000 | 0.0003387 | 0.0003275 | 0.0003355 | 0.0000079 |
| 9.0000000 | 0.0001241 | 0.0001200 | 0.0001234 | 0.0000034 |
| 10.0000000 | 0.0000455 | 0.0000440 | 0.0000454 | 0.0000014 |

*INPUT DATA SET USED*

H = 1.0000000

ICON = 1

**RESULTS OF ADAM4 PREDICTOR-CORRECTOR METHOD**

| X | YP | YC | YEXACT | ERROR |
|---|---|---|---|---|
| 0.0000000 | | | 1.0000000 | |
| 1.0000000 | 0.3750000 | | 0.3678794 | -0.0071206 |
| 2.0000000 | 0.1406250 | | 0.1353353 | -0.0052897 |
| 3.0000000 | 0.0527344 | | 0.0497871 | -0.0029473 |
| 4.0000000 | 0.0744628 | 0.0149522 | 0.0183156 | 0.0033634 |
| 5.0000000 | 0.0091045 | -0.0007950 | 0.0067379 | 0.0075330 |
| 6.0000000 | 0.0319238 | -0.0004662 | 0.0024788 | 0.0029450 |
| 7.0000000 | -0.0306429 | -0.0027268 | 0.0009119 | 0.0036387 |
| 8.0000000 | 0.0368799 | 0.0015700 | 0.0003355 | -0.0012345 |
| 9.0000000 | -0.0442679 | -0.0027738 | 0.0001234 | 0.0028972 |
| 10.0000000 | 0.0550162 | 0.0028112 | 0.0000454 | -0.0027658 |

## REFERENCES


1. McCalla, T. R., Introduction to Numerical Methods and Fortran Programming, John Wiley and Sons, Inc., New York, 1967.

2. Lapidus, L. and Seinfeld, J. H., Numerical Solutions of Ordinary Differential Equations, Academic Press, New York, 1971.

3. Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill Book Company, Inc., New York, 1956.

4. Hamming, R. W., Stable Predictor-Corrector Methods for Ordinary Differential Equations, J. ACM, Volume 6, 1959.

5. Henrici, P., Elements of Numerical Analysis, John Wiley and Sons, Inc., New York, 1964.

6. Ralston, A., A First Course in Numerical Analysis, McGraw-Hill Book Company, Inc., New York, 1965.

7. Chase, P. E., Stability Properties of Predictor-Corrector Methods for Ordinary Differential Equations, J. ACM, Volume 9, 1962.

8. Hull, T. E. and Creemer, A. L., Efficiency of Predictor-Corrector Procedures, J. ACM, Volume 10, 1962.

9. Crane, R. L. and Klopfenstein, R. W., A Predictor-Corrector Algorithm with an Increased Range of Absolute Stability, J. ACM, Volume 12, 1965.

# INITIAL DISTRIBUTION LIST

| | | No. Copies |
|---|---|---|
| 1. | Defense Documentation Center<br>Cameron Station<br>Alexandria, Virginia 22314 | 2 |
| 2. | Library, Code 0212<br>Naval Postgraduate School<br>Monterey, California 93940 | 2 |
| 3. | Assoc. Professor Uno R. Kodres, Code 53Kr<br>Department of Mathematics<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 4. | Philippine Military Academy<br>Fort Del Pilar, Baquio City<br>Philippines | 1 |
| 5. | Headquarters Philippine Navy<br>Roxas Boulevard, Manila<br>Philippines | 1 |
| 6. | Ltjg Romeo L. Malig,PN<br>6-B Tindalo, Project 3<br>Quezon City, Philippines | 1 |

UNCLASSIFIED

Security Classification

**DOCUMENT CONTROL DATA - R & D**

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

| | |
|---|---|
| 1. ORIGINATING ACTIVITY (Corporate author)<br>Naval Postgraduate School<br>Monterey, California 93940 | 2a. REPORT SECURITY CLASSIFICATION<br>Unclassified<br>2b. GROUP |

3. REPORT TITLE

COMPARISON OF PREDICTOR-CORRECTOR METHODS

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; June 1972

5. AUTHOR(S) (First name, middle initial, last name)

Romeo Liwanag Malig; Lieutenant Junior Grade, Philippine Navy

| 6. REPORT DATE | 7a. TOTAL NO. OF PAGES | 7b. NO. OF REFS |
|---|---|---|
| June 1972 | 204 | 9 |

| | |
|---|---|
| 8a. CONTRACT OR GRANT NO. | 9a. ORIGINATOR'S REPORT NUMBER(S) |
| b. PROJECT NO. | |
| c. | 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) |
| d. | |

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

| 11. SUPPLEMENTARY NOTES | 12. SPONSORING MILITARY ACTIVITY |
|---|---|
| | Naval Postgraduate School<br>Monterey, California 93940 |

13. ABSTRACT

The aim of this paper is to provide convenient predictor-corrector (P-C) methods for obtaining accurate numerical solution at a minimum cost to first order ordinary differential equations (ODE). In pursuing this goal, a unified development of the most popular and efficient P-C methods is presented, which includes derivation of formulas and analysis of error propagation and numerical stability. Each method is then coded and programmed using the Fortran language. Comparative analysis of the different P-C methods include both theoretical and numerical results. The numerical results were obtained by subjecting each method to a wide variety of test ODE, using a maximum of two corrector applications and a uniform series of step size values. By systematic comparison of the performance of each P-C method the most convenient P-C sets in terms of accuracy and minimum cost are established.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811



UNCLASSIFIED

Security Classification

A-31408

| 14. KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| EULER PREDICTOR-CORRECTOR METHOD | | | | | | |
| MILNE PREDICTOR-CORRECTOR METHOD | | | | | | |
| NYSTROM PREDICTOR-CORRECTOR METHOD | | | | | | |
| HAMMING PREDICTOR-CORRECTOR METHOD | | | | | | |
| HERMITE PREDICTOR-CORRECTOR METHOD | | | | | | |
| SECOND ORDER ADAMS PREDICTOR-CORRECTOR METHOD | | | | | | |
| THIRD ORDER ADAMS PREDICTOR-CORRECTOR METHOD | | | | | | |
| FOURTH ORDER ADAMS PREDICTOR-CORRECTOR METHOD | | | | | | |
| NUMERICAL STABILITY | | | | | | |
| ERROR PROPAGATION | | | | | | |
| CONVERGENCE | | | | | | |

Thesis
M27823 Malig 135112
c.1 Comparison of predictor-corrector methods.